HAL
open science

Environnements interactifs d'apprentissage et démonstration en géométrie

Dominique Py

To cite this version:

Dominique Py. Environnements interactifs d'apprentissage et démonstration en géométrie. 2001. hal-00190202

HAL Id: hal-00190202

https://telearn.hal.science/hal-00190202

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HABILITATION À DIRIGER DES RECHERCHES

présentée devant

L'Université de Rennes 1
Institut de Formation Supérieure
en Informatique et en Communication

par

Dominique Py

Environnements interactifs d'apprentissage et démonstration en géométrie

soutenue le 5 juillet 2001 devant le jury composé de

M.	Quinton	Président
M.	Balacheff	Rapporteur
Mme	Grandbastien	Rapporteur
Mme	Pastre	Rapporteur
M.	Besnard	Examinateur

Aux élèves de collège et à leurs enseignants de mathématiques,
qui m'ont tellement appris.

Table des matières

1 Introduction **5**

2 L'expert du domaine **7**
2.1 Représentation des connaissances géométriques 8
2.1.1 Objets et propriétés géométriques 8
2.1.2 Théorèmes et définitions 9
2.1.3 Énoncés et consignes 10
2.2 Démonstration 11
2.2.1 Etablir l'ensemble des faits et des pas 11
2.2.2 Etablir l'ensemble des démonstrations 13
2.3 Représentation dynamique des droites 14
2.4 Ajout d'objets non cités dans l'énoncé 17
2.5 Démonstration et explication 18

3 L'interaction **21**
3.1 Préparation de la session 22
3.2 Lecture de l'énoncé 22
3.3 Exploration du problème 23
3.4 Elaboration de la démonstration 24
3.5 Rédaction de la preuve 25

4 L'élève **27**
4.1 Résolution de problème et planification 28
4.1.1 Reconnaissance de plans 28
4.1.2 Plans et actions 29
4.1.3 Changements de plan 30
4.2 Diagnostic des erreurs 31
4.2.1 Utilisation d'un catalogue d'erreurs 31
4.2.2 Prise en compte du contexte 32
4.2.3 Exemple 33
4.3 Vers un modèle de la formation des erreurs 34
4.3.1 Modélisation de la tâche 34
4.3.2 Génération automatique de réponses 35
4.3.3 Exemple 36

4.3.4 Application au diagnostic 37

5 La classe **39**
5.1 Consignes pédagogiques et règles implicites 39
5.2 Réalisation et développements 42
5.3 Validation et expérimentations en classe 43
5.4 Utilisation dans l'enseignement 45
5.5 Interopération entre EIAOs 46

6 Conclusion **47**

A Chronologie succincte **55**

B Théorèmes du démonstrateur **57**

C Base d'exercices **61**

Chapitre 1

Introduction

Les recherches en EIAO (Enseignement Intelligemment Assisté par Ordinateur), initiées dans les années 70, se sont tout d'abord focalisées sur trois sujets, considérés par la plupart des auteurs comme les piliers des tuteurs intelligents :

- l'expert : un tuteur intelligent doit posséder des connaissances sur la discipline qu'il enseigne et savoir résoudre des problèmes.
- l'apprenant : un tuteur intelligent doit construire et faire évoluer un modèle individuel de l'élève auquel il s'adresse.
- le pédagogue : un tuteur intelligent doit disposer de stratégies pédagogiques afin de gérer les sessions d'enseignement.

A la fin des années 80, une autre déclinaison est apparue pour le sigle EIAO : Environnements Interactifs d'Apprentissage avec Ordinateur. Cette nouvelle interprétation traduit une évolution des recherches qui se placent résolument dans une perspective constructiviste et mettent l'accent sur l'interaction élève(s)/machine comme facteur d'apprentissage. Parallèlement, l'arrivée des micro-ordinateurs en milieu scolaire facilite l'usage des logiciels pédagogiques et donne lieu à des études sur l'impact de ces logiciels sur l'apprentissage et les pratiques d'enseignement [Artigue 97]. Enfin, le sigle EIAH (Environnements Informatiques d'Apprentissage Humain) a fait son apparition plus récemment.

L'EIAO est un domaine pluridisciplinaire qui conjugue psychologie cognitive, sciences de l'éducation, didactique, ergonomie, informatique et intelligence artificielle. Cette dernière y trouve un champ d'application très diversifié. L'expertise du domaine requiert des techniques de résolution de problèmes, de raisonnement ou de démonstration; la modélisation de l'élève fait appel à des méthodes de diagnostic, de reconnaissance de plan ou d'apprentissage; les stratégies pédagogiques sont généralement réalisées sous forme de systèmes à base de règles; enfin les différents formalismes de représentation des connaissances sont mis à contribution de manière transversale dans les tuteurs.

Parmi les disciplines qui ont donné lieu à des logiciels d'enseignement, qu'il s'agisse de tuteurs ou de micromondes, la géométrie tient une place importante. D'une part, c'est un domaine mathématique qui se prête relativement bien à une représentation informatique, et qui a inspiré très tôt des travaux sur la construction de figures ou la démonstration automatique. D'autre part, elle est liée dans le cursus scolaire à l'apprentissage de la preuve, tâche

réputée difficile et qui a suscité de nombreuses recherches dans le champ de la didactique et de la pédagogie.

Le projet Mentoniezh (*géométrie*, en breton) a débuté à l'Irisa de Rennes, au milieu des années 1980, à l'initiative d'enseignants de collège et de chercheurs en intelligence artificielle [Trilling 96]. Son objectif était de construire un logiciel complet d'aide à la démonstration, organisé autour de quatre phases : l'acquisition de la figure, la transformation de la figure, l'exploration des propriétés et l'exposé de la démonstration. Le projet initial a donné lieu à différents travaux réalisés à Rennes (Mentoniezh [Py 96b], Défi [Baulac et Giorgiutti 91]) ou à Grenoble (TALC [Desmoulins 94], GéoSpécif [Bouhineau 97]).

Ce document présente une synthèse des travaux que j'ai effectués dans le domaine de l'EIAO, au sein du tuteur *Mentoniezh*. Il est divisé en quatre parties correspondant aux grands thèmes de ce champ de recherches. Le premier chapitre est consacré à la modélisation du domaine enseigné et à l'expert de ce domaine. Le deuxième chapitre traite de l'interaction entre l'élève et la machine. Le troisième chapitre évoque la problématique de la modélisation de l'élève et de l'interprétation de son comportement. Enfin, le quatrième chapitre présente la mise en œuvre et les expérimentations en milieu scolaire.

Chapitre 2

L'expert du domaine

Pour construire des systèmes d'enseignement « intelligents », il est indispensable d'y intégrer des connaissances sur le domaine enseigné. Lorsque le tuteur doit simplement présenter les connaissances à l'élève, celles-ci sont représentées sous forme déclarative, mais lorsque le système doit aussi permettre à l'élève de résoudre des problèmes, il devient nécessaire d'y adjoindre des connaissances procédurales, règles ou stratégies.

Ce corpus de connaissances expertes est exploité de diverses manières. Il permet au tuteur :

- de résoudre les problèmes posés sans faire appel à un expert humain,
- de valider les réponses de l'élève par comparaison avec les réponses correctes, voire de les analyser plus finement en distinguant les éléments corrects des éléments erronés,
- de présenter des exemples de résolution de manière dynamique et interactive, pour encourager l'observation et l'imitation,
- de générer de nouveaux problèmes.

Par comparaison avec les systèmes experts classiques, les experts conçus pour l'EIAO présentent des spécificités : on parle de « résolveurs pédagogiques ». D'une part, ils doivent produire un résultat intelligible par l'élève, et pour cela utiliser des moyens (théorèmes, modes de déduction) susceptibles d'être présentés à l'élève, soit directement, soit après reformulation. D'autre part, lorsqu'un problème admet plusieurs solutions, ils doivent être capables de les générer ou du moins de les reconnaître, en particulier pour valider les réponses de l'élève.

Sur le domaine de la géométrie, un tuteur intelligent utilise, outre une représentation des connaissances géométriques élémentaires, un résolveur pédagogique qui prend la forme d'un démonstrateur. Ce résolveur peut viser différents objectifs :

- produire une bonne solution, comme le ferait un « parfait élève »,
- produire un ensemble de solutions, comme le ferait un enseignant qui prépare un problème,
- produire une trace de résolution, avec des essais et des échecs, comme le ferait un enseignant qui montre l'exemple.

Dans le cadre du tuteur *Mentoniezh*, nous avons développé ces deux derniers objectifs : un résolveur qui calcule un ensemble de solutions, afin de pouvoir contrôler le travail de l'élève, et un résolveur qui recherche une solution en expliquant sa méthode. On ne s'intéresse ici qu'aux démontrations déductives directes, qui représentent l'essentiel des démonstrations du niveau collège, en laissant de côté les démonstrations par l'absure ou par récurrence.

Dans ce chapitre, nous décrivons le résolveur de base du tuteur *Mentoniezh* ainsi que les extensions qui lui ont été apportées. Le premier paragraphe expose la représentation des connaissances géométriques : objets, propriétés, théorèmes. Le second paragraphe décrit le fonctionnement du démonstrateur. Les deux paragraphes suivants présentent des prolongements à ce démonstrateur de base : une représentation dynamique des droites, et l'introduction d'objets non cités dans l'énoncé. Enfin, le cinquième paragraphe décrit un « expert montrant l'exemple » développé à partir du démonstrateur.

2.1 Représentation des connaissances géométriques

Le langage de représentation des connaissances, appelé HDL pour *Hypotheses Description Language*, est un langage logique du premier ordre. La logique s'avère un formalisme bien adapté à l'expression de connaissances géométriques dans un résolveur. Elle permet une description des énoncés proche du langage naturel et facilite la traduction dans un langage de programmation logique tel que Prolog, le langage retenu pour la réalisation.

2.1.1 Objets et propriétés géométriques

Les principaux objets manipulés en géométrie élémentaire sont les points, les segments, les distances, les droites, les cercles, les triangles et les quadrilatères. Les objets complexes peuvent être désignés à l'aide de plusieurs objets de base : ainsi un segment peut être représenté par ses deux extrémités, une distance par un segment, un triangle ou un quadrilatère par la liste de ses sommets. C'est pourquoi nous ne retenons que trois objets de base, le point, la droite et le cercle. Le domaine couvert par ce langage inclut la géométrie affine, ainsi que les cercles et les propriétés liées à l'orthogonalité (droites perpendiculaires, triangle rectangle, médiatrice et rectangle). En revanche, il ne recouvre pas tous les rapports métriques, mais seulement les plus fréquents, **un** (distance égale) et **un demi** (moitié).

Le langage HDL est défini par :

- un ensemble de constantes {A, B, ... Z},
- un ensemble de variables {a, b, ... z},
- un ensemble de types : {point, droite, cercle} tels que toute constante ou variable possède un type et un seul,
- une fonction *droite* : point $\times$ point $\rightarrow$ droite,
- un ensemble de prédicats d'arité 2 : {appartient-droite, appartient-cercle, centre-cercle, parallèle, perpendiculaire, différent}.
- un ensemble de prédicats d'arité 3 : {alignés, milieu, diamètre, médiatrice, triangle-rectangle}.

– un ensemble de prédicats d'arité 4 : {égale, moitié, parallélogramme, losange, rectangle, convexe}.

La signification des prédicats est la suivante :

alignés(a,b,c)	Les points a, b, c sont alignés
milieu(a,b,c)	Le point a est le milieu de [bc]
appartient-cercle(a,c)	Le point a appartient au cercle c
centre-cercle(a,c)	Le point a est le centre du cercle c
diamètre(a,b,c)	Le segment [ab] est un diamètre du cercle c
égale(a,b,c,d)	La distance ab est égale à la distance cd
moitié(a,b,c,d)	La distance ab vaut la moitié de la distance cd
médiatrice(d,a,b)	La droite d est la médiatrice de [ab]
parallèle(a,b)	Les droites a et b sont parallèles
perpendiculaire(a,b)	Les droites a et b sont perpendiculaires
triangle-rectangle(a,b,c)	Le triangle abc est rectangle en a
parallélogramme(a,b,c,d)	Le quadrilatère abcd est un parallélogramme
losange(a,b,c,d)	Le quadrilatère abcd est un losange
rectangle(a,b,c,d)	Le quadrilatère abcd est un rectangle
appartient-droite(a,b)	Le point a appartient à la droite b
convexe(a,b,c,d)	Le quadrilatère abcd est convexe
différent(a,b)	Les objets a et b sont distincts

Forme canonique des termes

Les termes HDL sont mis sous forme canonique afin de réduire l'explosion combinatoire due aux nombreuses symétries de la géométrie. Lorsque deux objets jouent un rôle symétrique dans un terme HDL, ils sont placés dans l'ordre alphabétique. Ainsi, les faits *I est le milieu de [AB]* et *I est le milieu de [BA]* sont représentés par un seul et même terme, *milieu(I,A,B)*, puisque A et B jouent des rôles symétriques. Le terme *losange(A,B,C,D)* désigne indifféremment les losanges ABCD, BCDA, CDAB, DABC, DCBA, CBAD, BADC ou ADCB.

2.1.2 Théorèmes et définitions

Les règles du démonstrateur correspondent aux théorèmes et définitions du programme des classes de quatrième. Ces théorèmes et définitions sont répartis en dix grands groupes :

– composition du parallélisme et de l'orthogonalité,
– alignement de trois points,
– propriétés du cercle,
– transitivité de l'égalité (pour les distances),
– définitions et propriétés de la médiatrice,
– définitions et propriétés du triangle rectangle,
– définitions et propriétés du parallélogramme,
– définitions et propriétés du rectangle,

– définitions et propriétés du losange,
– théorème des milieux (Thalès) et sa réciproque.

Chaque théorème ou définition se traduit par une ou plusieurs clauses de Horn définies, telles que toute variable apparaissant en partie droite apparaît aussi en partie gauche. Par exemple, la définition

« Un triangle qui a deux côtés perpendiculaires est un triangle rectangle »

est traduite par deux clauses :

$\forall a,b,c,$ triangle-rectangle(a,b,c) $\Rightarrow$ perpendiculaire(droite(a,b),droite(a,c))

$\forall a,b,c,x,y,$ app-droite(a,x),app-droite(b,x),app-droite(a,y),app-droite(c,y), différent(a,b),différent(a,c),perpendiculaire(x,y) $\Rightarrow$ triangle-rectangle(a,b,c)

2.1.3 Énoncés et consignes

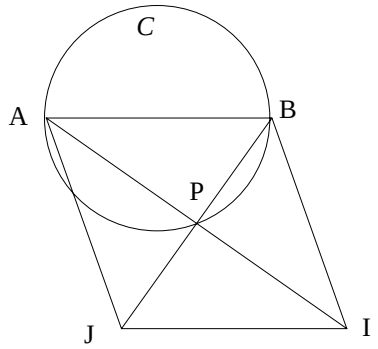
Un **énoncé** est constitué d'une conjonction de termes HDL appelés Données et d'un terme HDL appelé But. Par exemple, l'énoncé :

Soit C un cercle de diamètre [AB]. P est un point quelconque de C. Les points I et J sont tels que P est le milieu de [AI] et de [BJ]. Démontrer que le quadrilatère ABIJ est un losange.

est représenté par :

Données convexe(A,B,I,J), app-droite(A,dr1), app-droite(I,dr1), app-droite(P,dr1), app-droite(B,dr2), app-droite(J,dr2), app-droite(P,dr2), diamètre(A,B,G), appartient-cercle(P,C), milieu(P,A,I), milieu(P,B,J).

But losange(A,B,I,J).



Une base de quarante exercices a été constituée à partir d'énoncés provenant de manuels scolaires. Ces exercices, de difficulté variable, se répartissent en trois domaines : droites et cercles, quadrilatères, triangles. Cette base est utilisée comme jeu de test pour le démonstrateur.

Les **consignes** sont propres à une session de résolution. Elles partitionnent l'ensemble des théorèmes en trois sous-ensembles : les théorèmes cachés, les théorèmes explicites et

les théorèmes implicites. Les théorèmes « cachés » ne seront pas utilisés durant la démonstration. Les théorèmes « implicites » seront utilisés mais seront sous-entendus dans les preuves solutions.

2.2 Démonstration

La démonstration vise deux objectifs : établir l'ensemble des faits démontrables et des pas de preuves licites, recenser l'ensemble des preuves. Ces objectifs sont réalisés au cours de deux étapes successives.

2.2.1 Etablir l'ensemble des faits et des pas

Principe

En géométrie élémentaire, effectuer un pas de démonstration consiste à sélectionner un sous-ensemble de faits établis, puis un théorème dont les prémisses sont unifiables avec ces faits, et à déduire la conclusion de ce théorème. De manière plus formelle, si :

- P est un ensemble de littéraux positifs clos
- $t \in T$ est une clause de la forme $p_1,p_2,...p_n \Rightarrow q$
- il existe une substitution σ telle que $(p_1,p_2,...p_n)_\sigma \in P$

alors

on déduit $(q)_\sigma$

On appelle *pas de preuve* le triplet $< (p_1,p_2,...p_n)_\sigma,t,(q)_\sigma >$

Soit P_i un ensemble de littéraux, T l'ensemble des théorèmes, on note Q_i l'ensemble des littéraux immédiatement déductibles de P_i et de T.

Pour établir l'ensemble des faits démontrables, on réitère des déductions élémentaires jusqu'à saturation.

Soit P_0 l'ensemble des données du problème.

1. On établit Q_0 à partir de P_0 et de T. On note P_1 l'union de P_0 et Q_0.
2. On établit Q_1 à partir de P_1 et de T. On note P_2 l'union de P_1 et Q_1.
3. ...

On s'arrête dès que $Q_i \subset P_i$: P_i représente alors l'ensemble des faits démontrables.

Algorithme

Un algorithme de démonstration découle assez naturellement de cette définition. Afin de ne pas refaire plusieurs fois un même pas de preuve, on définit deux structures gérées en file, Faits-étudiés et Faits-à-étudier. Chaque étape consiste à retirer le premier élément de Faits-à-étudier et à en déduire toutes les conclusions possibles (en utilisant si besoin des hypothèses annexes elles-mêmes déjà étudiées), puis à l'ajouter à Faits-étudiés. Les pas de preuve sont mémorisés sous forme de triplets {*hypothèses, théorème, conclusion*} dans une liste Pas-de-preuve. Chaque pas est étiqueté *implicite* ou *explicite* selon le statut du théorème tel qu'il est précisé dans les consignes.

```
Pas-de-preuve := {}
Faits-étudiés := {}
Faits-à-étudier := Données
Tant que Faits-à-étudier ≠ {} faire
  Fait-courant := premier(Faits-à-étudier)
  Pour tout Théorème ∈ Ensemble-Théorèmes faire
    Si applicable(Théorème, Fait-courant, Faits-étudiés) alors
      étiqueter le pas comme implicite ou explicite puis l'ajouter à Pas-de-preuve
      Conclusion := conclusion(Théorème, Fait-courant, Faits-étudiés)
      Si Conclusion ∉ (Faits-étudiés ∪ Faits-à-étudier) alors
        ajouter(Conclusion, Faits-à-étudier)
      Finsi
    Finsi
  Refaire
Refaire
```

Parmi les pas de preuve, ceux qu'on ne peut effectuer *qu'après* avoir démontré la conclusion présentent un intérêt particulier. Ils sont appelés pas *impossibles* (tous les autres étant des pas *possibles*), puisqu'ils ne sont pas réalisables dans le contexte de l'exercice. Pour recenser ces pas impossibles, l'algorithme est légèrement modifié. La conclusion du problème, lorsqu'elle est démontrée, n'est pas ajoutée à l'ensemble Faits-à-étudier. Lorsque cet ensemble est vide, on est assuré que tous les pas de preuve qui ont déjà été produits sont des pas *possibles* et que tous ceux qui le seront ultérieurement sont des pas *impossibles*. On ajoute alors la conclusion à l'ensemble Faits-à-étudier de manière à produire ces pas. L'algorithme devient :

```
Pas-de-preuve := {}
Faits-étudiés := {}
Faits-à-étudier := Données
Tant que Faits-à-étudier ≠ {} faire
  Fait-courant := premier(Faits-à-étudier)
  Pour tout Théorème ∈ Ensemble-Théorèmes faire
    Si applicable(Théorème, Fait-courant, Faits-étudiés) alors
      étiqueter le pas puis l'ajouter à Pas-de-preuve (possible)
      Conclusion := conclusion(Théorème, Fait-courant, Faits-étudiés)
      Si Conclusion ∉ (Faits-étudiés ∪ Faits-à-étudier ∪ {But}) alors
        ajouter(Conclusion, Faits-à-étudier)
      Finsi
    Finsi
  Refaire
Refaire
Faits-à-étudier := {But}
Tant que Faits-à-étudier ≠ {} faire
  Fait-courant := premier(Faits-à-étudier)
  Pour tout Théorème ∈ Ensemble-Théorèmes faire
```

Si applicable(Théorème, Fait-courant, Faits-étudiés) **alors**
étiqueter le pas puis l'ajouter à Pas-de-preuve *(impossible)*
Conclusion := conclusion(Théorème, Fait-courant, Faits-étudiés)
Si Conclusion $\notin$ (Faits-étudiés $\cup$ Faits-à-étudier) **alors**
ajouter(Conclusion, Faits-à-étudier)
Finsi
Finsi
Refaire
Refaire

2.2.2 Etablir l'ensemble des démonstrations

Principe

De manière générale, une démonstration en géométrie est une séquence de pas qui permet de démontrer la conclusion du problème à partir de ses données. Plus précisément, une démonstration est un ensemble *minimal* de pas vérifiant certaines conditions : un de ces pas doit démontrer la conclusion du problème, aucun de ces pas ne doit comporter en hypothèse la conclusion du problème, et toute hypothèse d'un de ces pas doit être soit une donnée du problème, soit la conclusion d'un autre pas.

Définition : soit un énoncé défini par un ensemble de données $D = \{d_1,...,d_d\}$ et un but b. Un ensemble de pas $P = \{p_1,...p_n\}$ où chaque pas est de la forme $p_i = (h_{i1},...h_{im},t_i,c_i)$ est une preuve si et seulement si :

- $\exists i$ tel que $c_i = b$ et $\forall h_{ij}$, $h_{ij} \neq b$
- $\forall h_{ij}$, soit $h_{ij} \in D$, soit $\exists k$ tel que $h_{ij} = c_k$
- $\forall i, P - \{p_i\}$ n'est plus une preuve

Pour recenser les preuves, une méthode consisterait à énumérer tous les sous-ensembles de l'ensemble des pas, puis à retenir ceux qui vérifient la définition ci-dessus. Cette méthode, de complexité exponentielle, n'est pas envisageable dans la pratique. Un solution plus efficace consiste à parcourir l'espace des branches de manière descendante à partir de la conclusion.

Algorithme

L'énumération des preuves est basée sur une recherche en profondeur itérative (*iterative deepening*). Cette méthode permet de simuler la recherche en largeur d'abord (*breadth-first*) à un coût moindre, puisque le graphe de recherche n'est pas mémorisé explicitement. On énumère toutes les preuves de taille un, puis de taille deux, etc., jusqu'à la taille maximale recherchée.

Ensemble-preuves := {}
Pour p depuis 1 juqu'à profondeur-max **faire**
Tant que il reste des preuves de taille $\leq$ p **faire**
démonstration(but, {}, 0, p, preuve)

si taille(preuve) = p **alors** ajouter(preuve,Ensemble-preuves)
Refaire
Refaire

procédure démonstration (conclusion, liste-ancêtres, prof-courante, prof-max, preuve)
Si but $\in$ Données **alors** succès; preuve:={}
Sinon si Il existe un pas *possible* P = (H,t,c)
tel que (c = conclusion) et (H $\cap$ liste-ancêtres = {}) **alors**
Si P est un pas *explicite* **alors**
prof-courante := prof-courante + 1
Finsi
Si prof-courante $\leq$ prof-max **alors**
Tant que il reste des sous-preuves **faire**
démonstration2(H, liste-ancêtres $\cup$ {conclusion}, prof-courante, prof-max, sous-preuves)
succès; preuve := {sous-preuves} $\cup$ {P}
Refaire
Finsi
Finsi

procédure démonstration2 (liste-conclusions, liste-ancêtres, prof-courante, prof-max, sous-preuves)
Pour tout conclusion de liste-conclusions **faire**
démonstration(conclusion, liste-ancêtres, prof-courante, prof-max, preuve)
Refaire
Si succès **alors** sous-preuves := $\bigcup$ preuve **finsi**

Parmi les pas de preuve *possibles*, ceux qui font partie d'une preuve présentent un intérêt particulier. Ils sont appelés pas *utiles*, tandis que ceux qui n'appartiennent à aucune preuve sont des pas *inutiles*. Le recensement des pas utiles est réalisé par un simple parcours de la liste des preuves.

Pour les problèmes présentant un nombre de preuves très élevé (une dizaine d'exercices de la base), on se limite à l'ensemble des p plus courtes preuves. Sur le jeu d'essai considéré, il apparaît que $p = 50$ donne un ensemble significatif de preuves [1]. Ainsi, on ne perd pas véritablement de démonstrations intéressantes car au-delà, les preuves deviennent artificiellement compliquées.

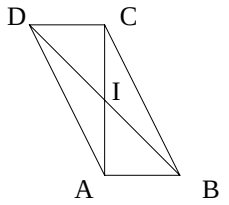
2.3 Représentation dynamique des droites

Une limite du langage HDL et du démonstrateur qui viennent d'être décrits réside dans la représentation des droites. En effet, celles-ci sont définies statiquement dans l'énoncé au moyen des prédicats *appartient-droite*, mais aucun théorème ne permet de démontrer qu'un

1. Cette valeur a été retenue sur des bases empiriques, elle pourrait varier si l'on considérait une autre classe de problèmes

point appartient à une droite, ou que deux droites sont confondues (il existe seulement des règles concluant sur le prédicat *alignés*). Cette limitation entraîne l'échec du démonstrateur sur certains problèmes. Considérons par exemple l'énoncé suivant :

> Soit ABCD un parallélogramme tel que (AB) est perpendiculaire à (AC). I est le milieu de [BD]. Montrer que AIB est un triangle rectangle en A.



Dans ce problème, l'alignement des points I, A et C résulte d'une déduction. Il n'est donc pas possible de montrer que ces points appartiennent à une même droite, et le démonstrateur échoue à établir le but.

Deux solutions classiques ne sont pas envisageables ici :

- Introduire les règles permettant de conclure sur *appartient-droite* et dupliquer les propriétés d'une droite pour chaque droite « synonyme ». Cette méthode produit une explosion combinatoire dès que le problème comporte des alignements de quatre points ou plus.
- Lorsque deux droites s'avèrent confondues, ne conserver que l'une des deux et lui attribuer toutes les propriétés de l'autres. Cette méthode amène à perdre des preuves, puisque l'on « oublie » des déductions antérieures.

Dans sa thèse, François-Gilles Carpentier propose de résoudre ce problème en introduisant un prédicat *confondues*, dont les deux arguments sont des droites, et en adaptant le mécanisme de la démonstration, de manière à éviter l'explosion combinatoire sans pour autant perdre de preuves intéressantes [Carpentier 98].

Pré-traitement de l'énoncé

Avant la démonstration proprement dite, l'énoncé est mis sous « forme normale ». Cette étape consiste à compléter l'énoncé en y introduisant les droites implicitement présentes. Ainsi, on recense :

- les alignements exprimés à l'aide des prédicats *milieu*, *alignés* et *appartient-droite*,
- les droites implicitement déclarées par des prédicats tels que *triangle-rectangle*, *parallélogramme*, *rectangle*, *losange* et *médiatrice*.

Pour chaque droite, on crée la constante associée et autant de prédicats *appartient-droite* que nécessaire. A ce stade, on traite l'égalité des droites en ne conservant qu'une des deux constantes et en substituant cette constante à toutes les occurrences de l'autre.

Principe

Au cours de la recherche, si l'on doit exprimer qu'une droite passant par les points *x* et *y* possède une propriété *p*, deux cas sont possibles.

S'il existe déjà une droite *d* tels que *x* et *y* appartiennent à cette droite, alors on produit le pas de déduction

appartient-droite(x,d), appartient-droite(y,d), ... $\Rightarrow$ p(d)

et on procède de même pour toutes les droites *d'* vérifiant *appartient-droite(x,d')* et *appartient-droite(y,d')*.

S'il n'existe pas de droite passant par *x* et *y*, alors on commence par créer une telle droite en introduisant les hypothèses *appartient-droite(x,d)* et *appartient-droite(y,d)*, puis on procède comme précédemment.

Alignement de points

Une propriété du milieu permet de conclure sur l'alignement de trois points :

Si I est le milieu de [AB], alors I, A et B sont alignés.

Le prédicat *alignés*, qui exprimait l'alignement de trois points dans la version initiale du démonstrateur, est rendu inutile par le nouveau mécanisme de gestion des droites.

S'il existe déjà une droite *d* tels que deux des trois points I, A, B appartiennent à cette droite, alors on produit les pas de déduction :

appartient-droite(x_i,d), appartient-droite(x_j,d), milieu(x_1, x_2, x_3) $\Rightarrow$ appartient-droite(x_k,d)

où (i,j,k) est une permutation de (1,2,3).

S'il n'existe pas une telle droite, alors on commence par créer une droite D en introduisant les hypothèses *appartient-droite(A,D)* et *appartient-droite(B,D)*, puis on procède comme précédemment.

Droites confondues

Le problème le plus délicat est celui du traitement de deux droites qui s'avèrent confondues, à la suite d'un pas de déduction tel que :

Si deux droites sont parallèles et ont un point commun, alors elles sont confondues

Si l'on se borne à introduire le prédicat *confondues(d1,d2)* et à dupliquer les propriétés de chacune des droites, on produit de nombreux faits et preuves redondants. Pour éviter cela, on introduit une nouvelle droite qui se substitue aux deux autres. Cependant, on souhaite ne pas perdre de preuve : il faut donc n'effectuer la substitution qu'après que toutes les déductions possibles aient été faites à partir des deux droites initiales. C'est pourquoi l'étude du prédicat *confondues* est retardé (de la même manière que l'étude de la conclusion du problème). Lorsque tous les autres faits de la liste Faits-à-étudier ont été considérés, on traite le prédicat *confondues(d1, d2)* en introduisant une nouvelle constante *d3* et en lui attribuant toutes les propriétés possédées par *d1* et *d2*. De plus, on masque désormais les constantes *d1* et *d2*, afin d'empêcher la déduction de faits tels que *confondues(d1, d3)*.

Conclusion

La gestion dynamique des droites qui vient d'être décrite permet de généraliser leur traitement et de manipuler des droites passant par un nombre quelconque de points. Le démonstrateur peut ainsi résoudre des problèmes sur lesquels il échouait auparavant, tout en conservant une complexité satisfaisante.

2.4 Ajout d'objets non cités dans l'énoncé

Une autre limite du démonstrateur réside dans le fait que tous les objets utilisés dans la démonstration doivent avoir été explicitement cités dans l'énoncé. Or l'introduction de nouveaux objets s'avère utile dans certains cas, soit que le problème ne puisse être résolu avec les objets de départ, soit qu'il existe une solution plus courte ou plus élégante se basant sur un nouvel objet.

Dans certains résolveurs, il est possible d'introduire un nouvel objet en cas d'échec, mais c'est à l'utilisateur que revient le choix de cet objet. Dans *Mentoniezh*, nous avons étendu le résolveur de manière à permettre l'ajout automatique d'objets, en se limitant aux objets de type *point* [Le Pape 97].

Principe

La procédure d'ajout est déclenchée lorsque la première phase de résolution (production des faits et des pas de preuve) échoue à démontrer la conclusion de l'exercice. A ce stade, on dispose d'une description relativement complète de la figure, qui facilite la recherche de sous-figures caractéristiques, lesquelles servent d'heuristiques pour le choix de l'objet à introduire.

Choix de l'objet

On se limite ici à l'ajout d'un seul objet à la fois. Pour choisir cet objet, on cherche à simuler, à l'aide de règles, le « coup d'œil » de l'expert qui repère sur la figure l'objet manquant. Deux types de règles heuristiques détectent des sous-figures pertinentes et l'objet manquant :

- Des règles basées sur le but du problème. Elles ajoutent un objet qui permettrait l'application d'un théorème démontrant la conclusion de l'exercice. Par exemple, lorsqu'on cherche à montrer qu'un quadrilatère est un parallélogramme, introduire le milieu de l'une de ses diagonales peut permettre d'appliquer la définition « Un quadrilatère dont les diagonales se coupent en leur milieu est un parallélogramme ».
- Des règles basées sur les données. Elles décrivent une sous-figure type, dont un élément est absent. Par exemple, si dans un triangle on connaît les milieux de deux côtés ainsi que les médianes correspondantes, alors on introduit le milieu du troisième côté.

Méthode

L'ajout d'objets se fait séquentiellement, après que la première tentative de démonstration ait échoué. A chaque étape :

- On recherche une règle de création applicable à la situation.
- Si toutes les règles ont déjà été essayées, on termine sur un échec.
- Sinon, on ajoute à l'énoncé l'objet indiqué par cette règle.
- On relance la démonstration sur l'énoncé complété.
- Si le but est prouvé, on termine sur un succès.
- Sinon, on efface l'objet ajouté ainsi que tous les pas de preuve qui en découlent.

Unicité des objets

L'ajout d'un objet peut amener à contredire le principe d'unicité des noms adopté dans *Mentoniezh*. Par exemple, on peut être amené à introduire le milieu X de la diagonale [AC] dans un parallélogramme ABCD alors que le milieu I de la diagonale [BD] est déjà présent : les points X et I sont alors confondus. Ce type d'ajout est parfois indispensable : dans certains exercices, pour prouver qu'un objet x possède une propriété P, il faut poser l'existence d'un objet vérifiant P, puis montrer que cet objet est nécessairement x. Il est donc utile d'autoriser ce type de situation, en adaptant la démonstration de manière à permettre la coexistence de deux points identiques, dans certains cas. Cette question est traitée de manière relativement similaire à celle des droites confondues.

La détection de la présence de deux points identiques est difficile. En effet, il n'existe pas de théorème scolaire explicite concluant sur ce type de fait (comme c'est le cas pour les droites). Certaines propriétés d'unicité peuvent cependant être utilisées pour formuler des règles de détection des doublons. Par exemple, si deux points possèdent la propriété d'être le milieu d'un même segment, on peut en déduire qu'ils sont identiques. Plusieurs règles de ce type ont été définies à partir des propriétés exprimables en HDL, sans que l'on puisse assurer qu'elles recouvrent tous les cas possibles. Ces règles ont pour conclusion un nouveau terme HDL *confondus* dont les deux arguments sont des points. Un méta-théorème *confondus*(p_1,p_2), $P(p_1) \Rightarrow P(p_2)$ permet d'attribuer à chaque point les propriétés de l'autre. Enfin, l'étude du prédicat *confondus* est retardé, et les constantes associées sont provisoirement masquées afin d'empêcher les déductions redondantes.

2.5 Démonstration et explication

En classe, pour enseigner la démonstration, le professeur peut commencer par effectuer lui-même une preuve afin de montrer l'exemple. De la même manière, les EIAOs utilisent souvent un résolveur capable d'expliquer son comportement pour présenter à l'apprenant un modèle de raisonnement expert. Dans *Mentoniezh*, la résolution telle que l'effectue le système ne peut être présentée comme exemple à un élève. En effet, l'approche exhaustive du démonstrateur, qui vise à recenser tous les faits, est à l'opposé de la pratique des experts humains. Si l'on veut présenter une trace de résolution à l'élève, il faut concevoir un expert de fonctionnement différent, qui simule l'activité de recherche d'un « professeur donnant

l'exemple », explore plusieurs pistes et rencontre des échecs. Cet expert produit une trace de son raisonnement, en justifiant ses choix à chaque étape [Turquety 93].

Principe

Le système, que l'on pourrait appeler « expert explorateur », cherche à produire un plan de preuve en parcourant l'espace du problème de manière descendante à partir du but. L'approche descendante permet de focaliser la recherche sur les hypothèses les plus pertinentes vis-à-vis du problème, contrairement à une approche purement ascendante qui ne tient pas compte de la conclusion à démontrer. Durant la recherche, l'expert humain utilise la figure comme heuristique, pour écarter les hypothèses non pertinentes. Ici, la figure est représentée par l'ensemble des faits déductibles produits par le démonstrateur au cours de la première phase de recherche. On adopte ici l'hypothèse du monde clos : tout fait non présent dans la base est considéré comme faux. Lorsqu'un but est décomposé en sous-buts dont l'un n'est pas vérifié sur la figure (c'est-à-dire qu'il n'a pas été déduit par le démonstrateur), le système conclut qu'il s'agit d'une impasse. Cependant, cette heuristique ne suffit pas à produire un raisonnement « naturel » ni à empêcher l'exploration de nombreuses pistes inutiles. D'autres heuristiques sont nécessaires pour que le système devienne efficace.

Ordonnancement des théorèmes

L'ordre dans lequel on cherche à appliquer les théorèmes pour établir un but donné a une grande influence sur le succès de la recherche. Il est indispensable de définir soigneusement un « ordre par défaut » des théorèmes, afin de limiter le nombre de retours-arrière effectués pendant le parcours. De manière générale :

- Les règles les plus spécifiques sont placées avant les plus générales. Ainsi, les théorèmes comportant des prémisses moins facilement satisfiables sont placés en tête.
- Les règles basées sur les mesures de longueurs sont défavorisées, car elles conduisent souvent à des sous-preuves très longues, faisant appel à la transitivité de l'égalité. Ainsi, les théorèmes comportant des prémisses de type *égale* ou *moitié* sont placés en dernier.

Sous-preuves immédiates (*look-ahead*)

L'ordre d'application des théorèmes subit des exceptions, lorsqu'un théorème non prioritaire permettrait de conclure plus rapidement sur le but courant. Ceci évite de s'engager dans une recherche fastidieuse alors que le résultat est à portée de main. Avant d'étudier les théorèmes selon l'ordre par défaut, le système examine si l'un d'eux permet de démontrer directement le but courant, et dans ce cas, il le place en tête de la liste.

Une fois le théorème choisi, il reste à déterminer son instanciation, qui n'est pas toujours unique. Dans le cas général, ces instanciations sont étudiées successivement dans un ordre quelconque. Cependant, pour quelques définitions des quadrilatères particuliers, le choix de l'instanciation est crucial car certaines conduiront à une sous-preuve très longue. Par conséquent, avant d'appliquer l'une de ces définitions, le système examine les différentes instanciations possibles et recherche si l'une d'elles correspond à des sous-buts dont

l'un est une donnée ou un fait déjà démontré. Dans ce cas, cette instanciation est étudiée en premier.

Mémorisation des succès et des échecs

Afin d'éviter des répétitions inutiles au cours de l'exploration, les propriétés « démontrées » sont mémorisées au fur et à mesure. Par exemple, pour montrer qu'un quadrilatère ABCD est un losange, on peut utiliser la définition :

Un parallélogramme dont les diagonales sont perpendiculaires est un losange

puis montrer que ABCD est un parallélogramme, mais échouer à montrer que ses diagonales sont perpendiculaires. On tente alors d'appliquer une seconde définition :

Un parallélogramme dont deux côtés adjacents sont égaux est un losange

On considère ici que l'on a déjà démontré que ABCD est un parallélogramme, et qu'il suffit de montrer l'égalité de deux côtés adjacents.

La mémorisation des échecs est plus délicate, car la notion d'échec sur un but est liée à l'emplacement de ce but dans l'arbre de preuve. Aussi, on limite la mémorisation des échecs à l'ensemble des sous-buts d'un but donné. Pour reprendre l'exemple précédent, si l'on cherche à montrer que ABCD est un losange avec la première définition, et que l'on échoue à montrer que ABCD est un parallélogramme, alors on ne tentera pas la seconde définition qui est vouée à l'échec puisque l'une de ses prémisses ne peut être démontrée.

Sous-preuves analogues

Dans un texte de démonstration, on rencontre fréquemment des ellipses telles que :« Sachant P_1,...P_n, on conclut C_1. De même, on conclut C_2 » où « de même » remplace « sachant Q_1,...Q_n » et où le théorème appliqué est identique dans les deux cas. Permettre ce type de formulation dans le système rend la trace de résolution plus concise et sa lecture plus agréable. On recherche donc les sous-buts frères de nature similaire, afin de tenter en priorité, pour démontrer le second, la règle qui a permis de démontrer le premier. Cette heuristique est utilisable à n'importe quel niveau de l'arbre de démonstration, mais ne porte que sur un pas de preuve.

Chapitre 3

L'interaction

La notion d'interaction tient désormais une place centrale dans la conception des environnements d'apprentissage. Alors que les premier travaux en EIAO s'attachaient à utiliser des techniques d'intelligence artificielle dans les logiciels d'EAO, afin d'en faire des systèmes à base de connaissances, il est progressivement apparu que le rôle de l'interaction dans les logiciels pédagogiques était fondamental. En effet, c'est la nature de l'interaction entre l'élève et la machine qui conditionne l'apprentissage, tout particulièrement en situation de résolution de problèmes.

Des travaux ont défini la notion de **situation d'interaction** afin de spécifier plus formellement l'interaction élève-machine [Delozanne 92]. D'autres recherches mettent l'accent sur le caractère adidactique des situations d'apprentissage qui peuvent être élaborées avec l'ordinateur [Luengo 97].

La géométrie élémentaire abordée au collège donne lieu à différentes activités autour de la figure (tracé, construction, manipulation) ou de la preuve (exploration, démonstration, rédaction). Cette variété de tâches papier-crayon se traduit par une variété de formes d'interaction dans les logiciels : il est nécessaire de définir un scénario d'interaction par type de tâche proposée à l'élève.

Classiquement, on distingue deux familles d'environnements d'apprentissage, en fonction de leur mode d'interaction avec l'élève : les tuteurs et les micromondes. Parmi les tuteurs de géométrie, on peut citer TALC (pour la figure), Geometry Tutor et Défi (pour la démonstration) [Desmoulins 94] [Anderson et al. 85] [Baulac et Giorgiutti 91]. Parmi les micromondes, citons Cabri-Géomètre et Sketchpad (pour la figure), Cabri-Euclide et Chypre (pour la figure et la démonstration) [Laborde 95] [Sutherland 63] [Luengo 97] [Bernat 94].

Mentoniezh se rattache plutôt à la famille des tuteurs, dans la mesure où le problème à résoudre est déterminé au préalable, et où il existe des solutions de référence (les preuves calculées par le démonstrateur). Cependant, l'activité de l'élève reste relativement libre, et aucune solution particulière ne lui est imposée. En ce sens, on peut donc considérer Mentoniezh comme un logiciel de type encadreur (ou *coach*) plutôt que comme un tuteur au sens strict.

Le domaine de Mentoniezh est celui de la démonstration (la figure n'est pas explici-

tement manipulée), depuis la lecture de l'énoncé de problème jusqu'à la rédaction de la preuve. La tâche de démonstration a été découpée en quatre sous-tâches, qui donnent lieu chacune à une étape de résolution suivant un mode d'interaction propre. Ces tâches sont la lecture de l'énoncé, l'exploration du problème, l'élaboration de la démonstration et la rédaction de la preuve. La progression de l'élève se fait de manière linéaire, chaque tâche n'étant abordée que lorsque la précédente a été validée. Contrairement à ce qui a été retenu dans d'autres tuteurs (comme Geometry Tutor), nous avons choisi de ne jamais fournir de solution partielle correcte à l'élève, ce qui nécessite en contrepartie d'apporter un soin particulier à la formulation des messages d'erreur.

Dans ce chapitre, nous décrivons les scénarios d'interaction correspondant aux quatre phases de résolution (lecture, exploration, démonstration et rédaction), en détaillant pour chacune d'elles ses objectifs, les actions que peut effectuer l'élève et les rétroactions du système.

3.1 Préparation de la session

Le problème est fourni au système sous la forme d'un fichier contenant l'énoncé en français, une liste d'objets typés (points, droites, cercles), une liste de faits HDL représentant les données et un fait HDL représentant la conclusion. Le démonstrateur produit alors la liste des faits démontrables, des pas de déduction et des preuves, comme décrit au chapitre précédent. C'est sur la base de ce résultat que va travailler le tuteur.

3.2 Lecture de l'énoncé

L'objectif de cette étape, pour l'élève, est de lire l'énoncé en français pour en extraire les propriétés mathématiques et les classer selon leur statut, *donnée* ou *conclusion*. L'élève peut aussi indiquer des propriétés qu'il lit sur la figure, elles ont alors le statut d'*observation*.

Au cours de cette phase, l'écran est divisé en trois fenêtres. La première contient l'énoncé du problème à résoudre, la seconde comporte une colonne **Données**, une colonne **Conclusion** et une colonne **Observations**, initialement vides, que l'élève doit compléter, et la troisième est une fenêtre de travail.

Les actions dont dispose l'élève sont :

- l'ajout d'une donnée, d'une conclusion ou d'une observation,
- le retrait d'une donnée, d'une conclusion ou d'une observation,
- la validation globale de son travail.

Pour ajouter un fait, l'élève indique le statut de ce fait puis choisit son type dans une liste de mots-clés correspondant aux termes HDL, et enfin il instancie le squelette de phrase avec les noms des objets (points, droites ou cercles). Par exemple, pour ajouter la donnée *I est le milieu de [AB]*, il sélectionne la commande *Ajouter une donnée*, puis le mot-clé *milieu* et enfin complète la phrase « *Le point . est le milieu de [..]* » avec les lettres I et AB.

Un premier contrôle est effectué à la saisie de chaque propriété. Le tuteur vérifie la syntaxe des objets cités (par exemple, un triangle doit être désigné par trois points distincts),

et s'assure qu'elle appartient bien aux propriétées recensées par le démonstrateur. Toute propriété non valide est refusée avec un message d'erreur.

Le contrôle global est effectué lorsque l'élève demande la validation. Le tuteur vérifie que la liste des données et la conclusion fournies par l'élève sont identiques à celles du problème. La seule exception concerne les alignements de quatre points ou plus. En effet, puisque le prédicat *alignés* du HDL comporte trois arguments, l'alignement de quatre points peut s'exprimer de plusieurs manières équivalentes à l'aide de deux termes HDL. Si un alignement de plus de trois points figure dans l'énoncé, le tuteur vérifie que les alignements indiqués par l'élève sont équivalents (mais non nécessairement identiques) à ceux de l'énoncé.

Si les propositions de l'élève sont correctes, le tuteur passe à l'étape suivante, sinon il indique le statut des éléments en surnombre ou manquants.

3.3 Exploration du problème

L'objectif de cette étape, pour l'élève, est d'étudier la figure et les sous-figures pertinentes afin d'esquisser une preuve, sans la détailler. Contrairement aux autres modules du tuteur, dans lesquels l'élève possède l'initiative, l'exploration se déroule sous la forme d'un dialogue dirigé par le tuteur [Py 00].

Au cours de cette phase, l'écran est divisé en deux fenêtres. La première affiche l'état de la démonstration (données et propriétés étudiées, propriétés en cours d'étude, propriétés à étudier), la seconde est une fenêtre de travail.

L'exploration est effectuée de manière descendante. Le but est décomposé en sous-buts (ou conditions suffisantes) de manière récursive, jusqu'à atteindre les hypothèses. Ceci amène à effectuer un parcours de l'espace de recherche sous-jacent au problème, avec des points de choix et des retours-arrière. Le parcours effectivement réalisé résulte de l'interaction entre le tuteur et l'élève. Chacun possède un rôle précis :

- le tuteur gère l'état du parcours, en maintenant à jour la liste des faits étudiés et leur statut,
- l'élève sélectionne, à chaque point de choix, le théorème à utiliser et son instanciation,
- le tuteur détecte les boucles et les impasses, il refuse les assertions fausses mais accepte les stratégies non optimales.

Les faits et leur évolution au fur et à mesure de la recherche sont gérés par le tuteur. Chaque fait peut avoir l'un des trois statuts suivants :

- faits étudiés : ce sont les données du problème et les faits pour lesquels l'élève a trouvé une preuve complète.
- faits à étudier : ce sont les faits qui n'ont pas encore été examinés par l'élève.
- faits en cours d'étude : ce sont les faits pour lesquels l'élève a proposé une décomposition partielle.

Lorsque l'élève ne sait pas quel théorème choisir parmi la liste proposée, il peut demander de l'aide. L'objectif de cette aide est d'amener l'élève à étudier chaque théorème relativement à la figure, de manière à voir s'il peut s'appliquer à la situation. Pour cela, le tuteur pose une série de questions, chaque question correspondant à l'un des théorèmes,

afin que l'élève se reporte à la figure, cherche la sous-figure liée au théorème et décide si le théorème est applicable. Ces questions sont de la forme « Vois-tu ... », la réponse attendue est *oui* ou *non*.

Si l'élève répond *non* à une question (que la réponse attendue soit *oui* ou *non*), le tuteur passe à la question suivante. Si l'élève répond *oui*, l'interaction continue exactement comme si l'élève avait sélectionné le théorème correspondant.

Si l'élève répond *non* à toutes les questions, il revient à son point de départ. Pour relancer l'exploration, le tuteur lui suggère alors de tracer sur sa feuille une sous-figure particulière, correspondant à une question à laquelle l'élève a répondu à tort par la négative.

L'exploration est terminée lorsque l'élève a réalisé le parcours complet de l'un des arbres solutions, c'est-à-dire qu'il a examiné tous les faits intermédiaires qui interviennent dans la preuve.

3.4 Elaboration de la démonstration

L'objectif de cette étape, pour l'élève, est de construire une démonstration pas à pas, en articulant hypothèses, théorème et conclusion. La démonstration peut être menée en chaînage avant (des hypothèses vers la conclusion), en chaînage arrière (de la conclusion vers les hypothèses) ou en chaînage mixte.

Au cours de cette phase, l'écran est divisé en trois fenêtres. La première affiche l'état de la démonstration (données et propriétés démontrées, propriétés à démontrer, conjectures), la seconde affiche l'état du pas de démonstration en cours (hypothèses, théorème, conclusion), et la troisième est une fenêtre de travail.

Les actions dont dispose l'élève sont :

- l'ajout d'une hypothèse, de la conclusion ou du théorème,
- le retrait d'une hypothèse, de la conclusion ou du théorème,
- la validation du pas courant,
- le retrait d'une conjecture.

Pour ajouter une hypothèse ou une conclusion, l'élève procède par sélection d'un mot-clé et instanciation d'une phrase-type, comme dans la phase de lecture. L'insertion d'un théorème se fait à partir d'un mot-clé à rechercher en prémisse ou en conclusion. Par exemple, une recherche sur le mot-clé *milieu* en conclusion fournira la liste des théorèmes permettant de démontrer qu'un point est le milieu d'un segment. L'élève parcourt ensuite cette liste et sélectionne l'un des théorèmes proposés.

Lorsque l'élève demande la validation du pas courant, le tuteur contrôle que ce pas est valide en le comparant à la liste des pas produits par le démonstrateur. Si le pas est correct, l'état de la démonstration est mis à jour. Deux cas sont possibles. Si toutes les hypothèses du pas sont des données ou des propriétés démontrées, la conclusion du pas devient elle-même une propriété démontrée. Si au moins une hypothèse du pas n'a pas été démontrée, alors la conclusion du pas devient une *conjecture*, et toutes les hypothèses non encore démontrées deviennent des buts.

L'étape de démonstration est terminée lorsque la conclusion du problème a été prouvée. Etant donné que le tuteur contrôle la validité de chaque pas, mais non sa « nécessité », cer-

taines propriétés peuvent avoir été démontrées inutilement, et il peut rester des conjectures non prouvées.

3.5 Rédaction de la preuve

L'objectif de cette étape, pour l'élève, est de rédiger un texte en français à partir de la preuve construite à l'étape précédente.

Dans un premier temps, si l'élève n'a pas réalisé sa preuve en chaînage avant, il doit remettre les pas de preuve dans l'ordre ascendant, en ne conservant que les déductions nécessaires. Puis, il compose sa rédaction phrase par phrase. L'écran est divisé en deux fenêtres : la première affiche l'état de la phrase en cours, la seconde affiche les différents éléments susceptibles de figurer dans la phrase (hypothèses, théorème, conclusion, mots de liaison, symboles de ponctuation).

Les actions dont dispose l'élève sont :

– l'ajout d'un élément,
– le retrait du dernier élément inséré,
– la validation de la phrase.

Lorsque l'élève demande la validation de la phrase courante, le tuteur effectue deux types de contrôle. Le premier contrôle porte sur la validité grammaticale de la phrase. A cet effet, une grammaire a été définie (via un automate d'états finis) pour exprimer les différentes tournures de phrases acceptées. Le second contrôle porte sur la validité logico-géométrique de la phrase et la présence des éléments mathématiques (hypothèses, théorème conclusion).

La session est terminée lorsque toutes les phrases ont été rédigées.

Les quatre tâches constituant l'activité de démonstration dans le tuteur ont donné lieu à des scénarios différents du point de vue de l'activité de l'élève, allant de la transcription d'un énoncé à l'ordonnancement de fragments de rédaction, en passant par l'exploration d'un espace de recherche et la construction logique d'une preuve. L'initiative reste à l'élève tout au long du travail, excepté dans la phase d'exploration où elle est partagée entre le logiciel et l'élève. La difficulté de réalisation informatique sous-jacente est variable, selon que le suivi du travail de l'élève fait appel à des techniques simples (lecture, rédaction) ou plus complexes (exploration, démonstration). Cependant, nous nous sommes attachés à conserver une homogénéité globale de l'environnement (menus, boîtes de dialogue), afin de faciliter la familiarisation de l'élève avec le logiciel.

Chapitre 4

L'élève

Les logiciels d'EAO sont, pour la plupart, destinés à être utilisés par un élève seul, voire deux élèves. Cette situation favorise une certaine individualisation de l'enseignement. L'élève progresse à son rythme, et lorsque le scénario pédagogique du logiciel n'est pas figé, il effectue un parcours qui lui est propre. Les concepteurs de tuteurs intelligents cherchent à renforcer cette caractéristique en intégrant au logiciel des capacités d'adaptation à l'élève, afin qu'il réagisse différemment selon les comportements manifestés par cet élève. Ceci est obtenu par l'élaboration et l'exploitation d'un « modèle de l'élève », ensemble d'informations propres à un apprenant. Ces informations peuvent être de natures diverses, mais le plus souvent elles portent sur les connaissances et les savoir-faire que le système attribue à l'élève au vu de son comportement.

La modélisation de l'élève soulève plusieurs problèmes complexes et encore mal résolus :

- Conception et modélisation : quelles caractéristiques représenter (connaissances, préférences, savoirs-faire, stratégies) et comment ?
- Initialisation : que contient le modèle avant le début de l'interaction ?
- Diagnostic : comment interpréter le comportement observé à l'interface ?
- Mise à jour, évolution : comment modifier le modèle lorsque les observations sont incompatibles avec l'état courant du modèle ?
- Usage : comment exploiter les informations recueillies ?

La démonstration en géométrie est une activité de résolution de problème, qui fait appel à des connaissances déclaratives ainsi qu'à des connaissances procédurales, des stratégies, des heuristiques. L'accent est mis sur ces dernières dans les modèles de l'apprenant conçus pour les tuteurs de géométrie. Les représentations usuelles sont basées sur les règles, et le diagnostic vise le suivi de la résolution (reconnaître les buts et les stratégies de l'élève) plutôt que l'évaluation de connaissances factuelles.

Dans le cadre du tuteur Mentoniezh, nous avons plus particulièrement étudié les questions concernant le diagnostic et l'interprétation du comportement de l'élève. Le diagnostic porte d'une part sur le niveau global de résolution du problème et d'autre part sur le niveau local de l'étape de résolution. Au niveau de la session, il s'agit de reconnaître le parcours

emprunté par l'élève afin d'anticiper sur ses actions. Au niveau du pas de résolution, il s'agit d'interpréter une entrée incorrecte afin d'élaborer une rétroaction adaptée. Par ailleurs, nous avons considéré un type de problème (exercices « à trous ») sensiblement différent de celui proposé dans Mentoniezh, afin d'étudier de manière plus précise le diagnostic des réponses erronées, ce qui nous a permis de définir un modèle simulant le comportement de l'élève.

4.1 Résolution de problème et planification

L'activité de résolution de problème peut être considérée comme une activité de planification : il s'agit de concevoir un plan puis de l'exécuter, avec des retours à la phase de planification en cas d'échec au cours de l'exécution. Suivre et analyser le travail d'un sujet en train de résoudre un problème requiert des connaissances sur les buts et les plans de ce sujet. Or, ceux-ci ne sont pas toujours explicites ni explicitables : il est nécessaire de les inférer à partir des actions effectuées par le sujet. Reconnaître les objectifs de l'utilisateur permet de lui apporter des aides ou des explications adaptées. Ainsi, plusieurs tuteurs sur le domaine des mathématiques ou de la programmation cherchent à inférer le but ou le plan de l'élève : Proust [Soloway et Johnson 85] compare la structure du programme écrit par l'étudiant à la structure des solutions, Macsyma Advisor [Genesereth 82] utilise l'analyse ascendante et la propagation de contraintes pour reconstruire le plan suivi par l'apprenant.

Mettre en œuvre des techniques d'inférence de plans ne signifie pas que l'on attribue à l'élève l'intention explicite de réaliser un plan donné. De manière générale, la reconnaissance de plan cherche davantage à modéliser le *raisonnement de l'observateur* (« Que conclure au vu de tel événement ? ») que celui du sujet observé. Le plan (ou la succession de plans) constitue une représentation synthétique et plausible du *comportement observé* à l'interface du système. Il n'est pas supposé représenter l'état mental ou cognitif du sujet à un instant donné. Il constitue une source d'information pour l'élaboration de rétroactions cohérentes avec l'activité de l'élève.

Dans le cas des problèmes de géométrie traités par Mentoniezh, l'objectif à atteindre - ici, une propriété à démontrer - est connu, puisqu'il est fourni dans l'énoncé. Cependant, les moyens d'y parvenir sont multiples : il peut exister plusieurs démonstrations et chacune peut être réalisée suivant différentes démarches. Il est donc intéressant de chercher à reconnaître le chemin emprunté par un élève particulier.

4.1.1 Reconnaissance de plans

La modélisation d'actions et de plans est un thème important de l'intelligence artificielle. Le terme « plan » connaît différentes acceptions, selon le contexte dans lequel il est utilisé. Nous retenons ici la définition la plus stricte, employée par les premiers systèmes de planification. Un **plan** est un enchaînement d'actions, *entièrement déterminé*, visant à atteindre un but donné, l'accomplissement de chaque action étant conditionné par l'effet des précédentes. La **génération de plans** consiste à déterminer la séquence d'actions produisant le but recherché. La **reconnaissance de plans** est le processus réciproque. Elle consiste à découvrir, à partir de l'observation de quelques actions, le plan mis en œuvre par le sujet observé.

Parmi les travaux réalisés dans ce domaine, ceux de H.A. Kautz [Kautz 87] définissent une théorie formelle de la reconnaissance de plans, qui s'appuie sur un modèle de raisonnement non monotone, la *circonscription*. Cette théorie s'applique à une base de connaissances (spécifique au domaine considéré) qui décrit les plans, les actions, et les relations logiques entre plans et actions. Elle repose sur un principe central de minimisation (ou d'économie), en vue de produire une **explication** aux observations. Une **explication** est définie ici comme un **ensemble non vide de plans** recouvrant toutes les observations, c'est-à-dire que toute action observée fait partie d'un plan de cet ensemble. Le principe de minimisation se justifie par le fait que les sujets observés cherchent à réaliser correctement leurs plans, et donc que les actions observées ont une certaine *cohérence*.

En pratique, les relations entre plans et actions donnent lieu à un ensemble d'axiomes, qui permettent de tirer formellement les conclusions d'une observation donnée. Cette théorie est décrite de manière détaillée dans [Kautz 87], et son application à Mentoniezh dans [Py 90].

4.1.2 Plans et actions

Dans le contexte du tuteur, réaliser une démonstration consiste à appliquer des théorèmes, soit à des données pour démontrer de nouvelles assertions, jusqu'à atteindre le but (c'est le *chaînage avant*), soit à un but pour le décomposer en sous-buts, jusqu'à atteindre les données (c'est le *chaînage arrière*), soit par une combinaison de ces deux méthodes (c'est le *chaînage mixte*). L'application d'un théorème constitue un pas de démonstration. Une démonstration est une suite de pas qui permet d'aboutir à une conclusion à partir d'un ensemble de données. Dans le formalisme des plans, on pose les correspondances suivantes : une action est un pas de démonstration, un plan est une démonstration.

Exemple Revenons à l'exemple du paragraphe 2.1.3 et considérons que ce problème ne comporte que deux preuves. La première preuve, **P1**, comprend les étapes **E1**, **E2** et **E3**, la seconde, **P2**, comprend **E1**, **E2**, **E4** et **E5** :

Preuve 1
(E1) P est le milieu de [AI] et [BJ], donc ABIJ est un parallélogramme
(E2) P appartient à C de diamètre [AB] donc (PA) $\perp$ (PB)
(E3) ABIJ est un parallélogramme et (AI) $\perp$ (BJ), donc ABIJ est un losange
Preuve 2
(E1) P est le milieu de [AI] et [BJ], donc ABIJ est un parallélogramme
(E2) P appartient à C de diamètre [AB] donc (PA) $\perp$ (PB)
(E4) P est le milieu de [AI] et (AI) $\perp$ (BP) donc (BP) est la médiatrice de [AI],
(E5) ABIJ est un parallélogramme et BA = BI , donc ABIJ est un losange

Supposons que nous observons des étapes successivement réalisées par un élève.

Première observation L'élève effectue l'étape E2. Ceci ne suffit pas à déterminer avec précision le plan mis en œuvre. Puisque E2 appartient à P1 et à P2, deux explications également plausibles apparaissent : {P1} et {P2}.

Seconde observation L'élève effectue E3. Puisque cette action appartient au plan P1 et à lui seul, on en déduit que le plan P1 est en cours d'exécution. Si les deux actions étaient sans lien, on pourrait considérer que E2 est expliquée par P2 et E3 par P1, ou bien que E2 et E3 sont toutes deux expliquées par P1. Mais le principe de minimisation écarte la première explication, et retient la seconde, puisqu'elle explique l'ensemble des observations par un seul plan. On conclut donc à l'explication unique {P1}. La suite logique des événements est l'action E1.

4.1.3 Changements de plan

Lorsque le sujet observé exécute un plan unique, la séquence d'observations entraîne une explication unique, ou une disjonction d'explications, chacune comportant *un seul* plan. Au fur et à mesure des observations, cette disjonction tend à se réduire à une explication unique, car les nouvelles actions écartent des explications devenues incohérentes. À l'opposé, lorsque le sujet change de plan, les explications produites comportent plusieurs plans, en même temps que leur nombre augmente.

La théorie de Kautz admet que le sujet observé peut mener plusieurs tâches de front : elle traite sur un pied d'égalité les différents plans qui figurent dans une explication, ainsi que les différentes explications d'une même séquence d'observations. Cette méthode, si elle est bien adaptée à la reconnaissance de plans se déroulant en parallèle, ne convient pas à la résolution de problèmes scolaires de géométrie. En effet, l'exécution d'un seul plan suffit à remplir l'objectif fixé : une preuve suffit pour démontrer la conclusion du problème. Mais le sujet peut commencer puis abandonner plusieurs plans avant de se fixer sur le plan définitif, ou bien reprendre un plan laissé en suspens. Il est donc nécessaire de pouvoir d'une part distinguer, parmi les plans possibles, un plan principal et des plans secondaires, d'autre part favoriser, parmi l'ensemble des explications produites, l'explication qui apparaît la plus plausible. Nous avons défini les notions de *plan principal* et de *plan courant*, ainsi que deux critères de préférence parmi les explications.

Le plan principal est celui qui regroupe le plus grand nombre d'observations (en cas d'ex-aequo, il existe donc plusieurs plans principaux). Tous les autres plans sont des plans secondaires. Le plan courant est celui qui recouvre l'observation la plus récente. Tous les autres sont des plans en attente. Si plusieurs plans sont en cours d'exécution, le plan principal et le plan courant ne sont pas nécessairement confondus. Cette distinction permet d'écarter les observations incohérentes.

Les critères de préférence, définis sous forme d'heuristiques, permettent de sélectionner une des explications en fonction du style de reconnaissance attendu. La première heuristique favorise le plan suivi le plus fréquemment. Elle retient l'explication dont le plan principal regroupe le plus grand nombre d'actions. La seconde heuristique favorise le plan suivi le plus récemment. Elle retient l'explication dont le plan courant regroupe le plus grand nombre d'actions. Ces deux heuristiques tendent à prédire des comportements différents : la première favorise un plan régulier, et tolère des données incohérentes ponctuelles ; la seconde favorise le dernier plan suivi, et reconnaît rapidement les changements de plan.

4.2 Diagnostic des erreurs

La question de l'analyse automatique des erreurs de l'élève se pose dès que l'on sort du cadre fermé des exercices de type QCM et que l'on cherche à dépasser le diagnostic binaire « réponse correcte/incorrecte » obtenu par comparaison de la réponse de l'élève avec la réponse attendue.

La prise en compte des erreurs dans les tuteurs intelligents se fait essentiellement de deux manières [Kass 87] : par l'exploitation de catalogues d'erreurs courantes, ou par la reconstruction des procédures erronées à partir de primitives. La première méthode requiert le recensement des erreurs les plus fréquentes, à partir de travaux d'élèves. Elle est simple à implémenter, mais son domaine de validité est lié au corpus de travaux étudié. La seconde se base sur une théorie de la formation des erreurs. Elle est plus générale, mais sa difficulté de mise en œuvre la limite à des domaines relativement simples (arithmétique, algèbre élémentaire).

Par ailleurs, la recherche en didactique s'intéresse depuis plusieurs années au rôle de l'erreur dans l'apprentissage. Celle-ci n'est plus considérée comme une faute ou une lacune, mais plutôt comme un révélateur des conceptions de l'élève. Il apparaît vain de chercher à prévenir ou à empêcher l'erreur : « si ces erreurs ne sont pas isolées mais témoignent de conceptions cohérentes, [...] il faut alors concevoir les moyens de déstabiliser ces conceptions, d'amener l'élève à comprendre son propre fonctionnement dans ce qu'il a de positif et de négatif à la fois.» [Artigue 92].

La démonstration en géométrie n'a pas encore donné lieu à l'élaboration d'une théorie automatisable de la formation des erreurs, et les approches usuelles se basent sur des catalogue d'erreurs [Anderson et al. 85] [Baulac et Giorgiutti 91]. A chaque théorème, ou règle correcte, est associé un ensemble de variantes, les règles incorrectes. Pour analyser la réponse de l'élève, on tente de l'apparier avec une règle correcte, puis, en cas d'échec, avec une règle incorrecte. Une rétroaction spécifique (message d'erreur, contre-exemple, correction partielle) peut être attachée à chaque règle incorrecte.

4.2.1 Utilisation d'un catalogue d'erreurs

Dans la première version de Mentoniezh, nous avons défini un catalogue d'erreurs de ce type, centré sur la **structure** du pas [Py 93]. Les erreurs sont réparties en trois classes :

Syntaxiques Il s'agit de l'utilisation d'un objet géométrique (point, droite, cercle) non présent dans l'énoncé, ou bien de la désignation d'un objet composé possédant un nombre de sommets incorrects ou deux sommets identiques.

Logiques Elles consistent en l'inversion de paramètres jouant un rôle symétrique. Par exemple, la permutation d'une hypothèse et de la conclusion, la confusion entre le théorème direct et le théorème réciproque, ou l'ordonnancement incorrect des sommets dans un quadrilatère.

Sémantiques Elles reposent sur une confusion de vocabulaire (parallèle et égale, milieu et moitié, droite et segment) entraînant une confusion entre les théorèmes.

Les erreurs de syntaxe sont détectées dès la saisie et donnent aussitôt lieu à un message d'erreur. Les deux autres catégories sont décrites dans la bibliothèque d'erreurs. A

chaque erreur est associée une méta-règle, composée d'un schéma de règle incorrecte et d'un schéma de règle réciproque, et une explication. Une méta-règle est applicable si le schéma incorrect s'apparie avec la réponse incorrecte de l'élève. Le diagnostic consiste à envisager toutes les méta-règles applicables et retenir celle dont la règle réciproque génère une réponse correcte. L'explication liée à cette métarègle constitue l'explication de l'erreur, fournie à l'élève. Lorsqu'aucune erreur n'a pu être reconnue, le processus est réitéré sur l'ensemble des résultats précédents, afin de reconnaître une éventuelle combinaison de deux erreurs (au-delà, le diagnostic devient hasardeux).

4.2.2 Prise en compte du contexte

Au cours des expérimentations en collège [Py et El Gass 92], il s'est avéré que les messages d'erreurs étaient parfois insuffisamment précis et ne permettaient pas à l'élève de comprendre quelle faute il avait commise. Cette imprécision ne tient pas à des lacunes de la bibliothèque d'erreurs - auquel cas il aurait suffi de la compléter - mais plutôt au principe même du diagnostic, qui considère uniquement la structure du pas de preuve, et non le contexte dans lequel il est réalisé. Or, en situation de résolution de problème, puisqu'il n'existe pas une réponse unique attendue, il est nécessaire de reconnaître en même temps l'erreur commise et la réponse visée par l'élève. Par exemple, si l'élève affirme (A et B représentent deux assertions) :« A, or $B => A$, donc B », deux interprétations sont possibles.

- L'élève a confondu théorème direct et théorème réciproque, il voulait dire : « A, or $A => B$, donc B ».

- L'élève a interverti hypothèses et conclusions, il voulait dire : « B, or $B => A$, donc A ».

On voit ici que le pas de preuve, considéré isolément, ne permet pas de conclure. Il faut faire appel au contexte pour départager les diagnostics. Si A est déjà démontré, et que B reste à prouver, on préfère le premier diagnostic, dans le cas contraire on préfère le second. Comme dans le cas de la reconnaissance de plan, il ne s'agit pas d'attribuer à l'élève l'intention explicite d'effectuer un pas de preuve donné, mais de trouver des critères pour produire un message d'erreur cohérent avec l'assertion de l'élève.

La fonction de diagnostic a donc été redéfinie, afin de prendre en compte le contexte de l'erreur. L'objectif est de produire un diagnostic qui soit d'une part plausible (le diagnostic idéal n'est pas réalisable) et d'autre part précis : un message tel que « Hypothèse *ou* théorème incohérent » est trop vague pour que l'élève retrouve son erreur par lui-même.

Un nouveau catalogue des erreurs a été établi, à partir des traces du travail des élèves (interactions avec l'ordinateur, brouillons et rédactions sur papier). Ces erreurs sont hiérarchisées, de manière à départager les diagnostics candidats, c'est-à-dire les différents couples {erreur, étape visée} qui expliquent une proposition erronée de l'élève. L'ordonnancement repose sur un certain nombre de principes directeurs, certains généraux à la résolution de problèmes, d'autres spécifiques à la démonstration. Tous reposent sur l'hypothèse selon laquelle on cherche en priorité à reconnaître une erreur mineure. Autrement dit, on suppose que la réponse correcte visée est la plus proche possible de la réponse incorrecte fournie. Cette heuristique est couramment retenue dans les tuteurs. En effet, prendre l'hypothèse inverse amènerait à reconnaître, dans toute proposition correcte, la combinaison de

plusieurs erreurs s'annulant mutuellement.

Les principes généraux sont les suivants : préférer une erreur unique à une combinaison d'erreurs, préférer une erreur spécifique à une erreur générale, préférer une étape utile dans la preuve à une étape inutile, et préférer une étape cohérente avec la résolution de l'élève. Les principes particuliers amènent, par exemple, à considèrer les erreurs portant sur les hypothèses avant les erreurs portant sur la conclusion, car ces dernières paraissent moins fréquentes au vu des expérimentations.

Le processus de diagnostic se déroule en trois étapes :

- On s'assure que les contraintes portant sur la structure du pas et de la preuve sont satisfaites. Par exemple, la conclusion ne doit pas figurer parmi les hypothèses, il ne doit pas exister de circuit de la forme « A parce que B et B parce que A ». Le message d'erreur pointe la contrainte non respectée.

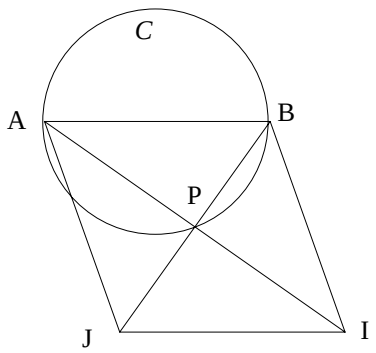
- On vérifie que le pas de l'élève figure parmi les différentes étapes correctes (fournies par le démonstrateur). Si ce n'est pas le cas, il faut rechercher le pas le plus similaire à celui proposé par l'élève, en fonction des critères définis ci-dessus. Pour guider cette recherche, on utilise le contexte, c'est-à-dire les données et les différentes solutions du problème, ainsi que les résultats de la reconnaissance de plan. Le message d'erreur pointe l'élément du pas (hypothèse, théorème ou conclusion) diagnostiqué comme incohérent.

- Il se peut néanmoins que la réponse de l'élève semble incohérente, relativement au problème, à tel point qu'aucun pas « voisin » ne puisse être trouvé. Dans ce cas, on examine si la conclusion visée est pertinente dans le cadre du problème et le message d'erreur incite l'élève à démontrer cette conclusion à l'aide d'autres prémisses.

Afin de valider ce nouvel algorithme de diagnostic, les sessions d'expérimentation ont été rejouées, pour comparer le comportement des deux méthodes de diagnostic dans des situations identiques. Cette simulation a permis de noter une amélioration notable de la précision des messages d'erreur.

4.2.3 Exemple

Revenons au problème du paragraphe 2.1.3.



L'élève propose le pas :

- Hypothèses : ABIJ est un parallélogramme et (AI) $\perp$ (BJ)
- Théorème : Si un parallélogramme a deux côtés adjacents égaux, alors c'est un losange
- Conclusion : ABIJ est un losange.

La seconde hypothèse ne concorde pas avec le théorème : ici, on attend soit une hypothèse de type *AJ = IJ*, soit le théorème *si un parallélogramme a ses diagonales perpendiculaires, alors c'est un losange*.

Sur cet exemple, quel que soit l'état d'avancement de la preuve, le diagnostic fondé sur la structure conclut seulement à une incohérence entre la seconde hypothèse et le théorème, sans plus de précision. Le diagnostic utilisant le contexte produit deux résultats différents, selon l'état d'avancement de la résolution. En début de session, on fait par défaut l'hypothèse que l'élève va réaliser la preuve la plus courte, en trois pas. Le pas incorrect est rapproché du dernier pas de cette preuve, dont il ne diffère que par le théorème, et le message d'erreur porte sur le choix de ce théorème. En revanche, si l'élève vient de démontrer que *AJ = IJ*, on lui attribue l'intention (grâce à la reconnaissance de plan) de s'engager dans une preuve plus longue, consistant à montrer que deux côtés adjacents sont égaux. Le pas incorrect est rapproché du dernier pas de cette preuve, dont il ne diffère que par la seconde hypothèse, et le message d'erreur porte sur le choix de cette hypothèse.

4.3 Vers un modèle de la formation des erreurs

Le diagnostic basé sur un liste d'erreurs courantes, s'il est relativement simple à mettre en œuvre, présente des inconvénients : ces listes sont coûteuses à établir, très liées au corpus étudié, et ne possèdent aucune valeur explicative. Ces limites ont amené des chercheurs à proposer des modèles qui reconstruisent les procédures erronées à partir de micro-perturbations sur des procédures correctes [Burton 82], ou des modèles génératifs, qui expliquent l'origine des erreurs [Van Lehn 88].

Actuellement, aucun modèle de ce type n'a été proposée pour la preuve. La nature de la tâche, que l'on peut assimiler à une recherche heuristique dans un graphe et/ou, la rend plus difficile à appréhender que les calculs arithmétiques, assimilables à des algorithmes, ou les transformations algébriques élémentaires, assimilables à une recherche heuristique dans un graphe d'états. C'est pourquoi nous avons abordé ce problème par le biais d'une tâche plus simple, la démonstration à un pas. L'exercice consiste pour l'élève à compléter un pas de preuve dont on a retiré une hypothèse, ou le théorème, ou encore la conclusion.

4.3.1 Modélisation de la tâche

Un test sur papier, comportant dix questions de ce type, a été élaboré et soumis à plusieurs classes (totalisant cent vingt élèves) afin de recueillir un échantillon significatif de réponses. L'analyse de ces questionnaires a permis de formuler l'hypothèse selon laquelle le raisonnement des élèves peut être simulé en utilisant le raisonnement par analogie et la satisfaction de contraintes [Py 94].

L'analogie est un mode de raisonnement qui met en correspondance des structures, afin de transposer des connaissances d'une situation à une autre [Haton et al. 91]. Elle se fonde sur la notion de similitude d'objets, de type *A est à B ce que C est à D*. Elle intervient en géométrie lorsque l'on cherche à résoudre un problème en se remémorant un exercice ou

une sous-figure similaire. Dans le cas de notre test, la tâche consiste à faire une analogie avec les théorèmes appris pour retrouver l'élément manquant. Si l'analogie porte sur des traits non pertinents, elle peut produire des résultats faux.

Les contraintes sont des relation entre deux ou plusieurs variables, qui restreignent les valeurs possibles pour ces variables. Les expressions géométriques (propriétés, théorèmes, déductions) sont soumises à un certain nombre de contraintes. Ces contraintes sont liées au sens des concepts géométriques en jeu. Par exemple, dans un pas de déduction, l'hypothèse doit être différente de la conclusion. Dans l'expression "x est le milieu de [yz]", les variables x, y, z doivent désigner des points et ces points doivent être tous distincts (si l'on écarte les cas limites).

Nous posons l'hypothèse que les réponses sont produites par l'interaction des analogies et des contraintes. Les analogies produisent des candidats solutions, acceptables ou non selon qu'elles se basent sur des traits pertinents du problème ou sur des traits de surface, et les contraintes éliminent ceux de ces candidats qui ne respectent pas les lois géométriques. Cette hypothèse est à la source d'un modèle de l'élève **exécutable** qui simule les réponses des élèves pour des exercices similaires au test.

4.3.2 Génération automatique de réponses

Le problème peut se définir ainsi : étant donné une question Q, générer *n* réponses R_1,R_2,...R_n plausibles (c.-à-d. que les élèves ont données ou auraient pu donner) ainsi que les explications correspondantes (c.-à-d. l'analogie utilisée, les contraintes respectées ou non).

Le modèle exploite cinq formes d'analogies, réparties en deux types principaux. Les analogies du premier type font appel uniquement à des données lisibles de manière immédiate (éléments d'une question voisine ou bien propriété de la figure). Ces analogies ne se combinent pas entre elles. Les analogies du second type font appel à la structure des théorèmes du cours et supposent que l'élève s'est remémoré le théorème utilisé. Ces analogies peuvent se combiner entre elles. Les contraintes représentées sont de cinq types : contraintes de typage, structurelles, contextuelles, syntaxiques et sémantiques.

Le simulateur se base sur une représentation des connaissances très proche de celui de Mentoniezh, sur un domaine un peu plus restreint. On considère :

- des objets géométriques (points, droites, cercles), chaque objet étant désigné par une lettre (A, B, C, ...).

- des propriétés géométriques exprimées par des prédicats dont les arguments sont des objets géométriques (parallèle, perpendiculaire, distances égales, points alignés, milieu, parallélogramme, losange, rectangle, triangle rectangle).

- des squelettes de théorème, au nombre de dix-huit, composés d'hypothèses (une à trois propriétés) et d'une conclusion (une propriété). Ces propriétés sont des prédicats **sans** leurs arguments.

On y adjoint un ensemble de sept règles d'instanciation. Ces règles expriment différentes manières de transformer un n-uplet d'arguments en un autre n-uplet d'arguments. Par exemple, la règle $4 \rightarrow 2 \times 2$ exprime les transformations d'un quadruplet en un doublet de doublets, comme : (ABCD) $\rightarrow$ (AB)(CD), (ABCD) $\rightarrow$ (AC)(BD), etc.

Une question est fournie au simulateur sous la forme d'un squelette de théorème incomplet (auquel il manque une hypothèse ou une conclusion), un n-uplet d'objets et une liste de propriétés décrivant la figure associée. Une réponse est produite sous la forme d'une propriété et d'un n-uplet d'objets.

Le système énumère, pour chaque forme d'analogie, les réponses que celle-ci peut produire. Les analogies du premier type sont simples à traiter, puisqu'il suffit de reprendre des propriétés citées dans la question ou dans la description de la figure associée. Les analogies du second type nécessitent de comparer la question avec la base de théorèmes et de règles d'instanciation. Pour cela, le générateur cherche tout d'abord à déterminer la propriété manquante dans la question, puis il instancie cette propriété, en respectant son arité. Pour trouver la propriété manquante, on recherche parmi la base de théorèmes celui ou ceux qui s'apparient avec le théorème de la question posée. Pour instancier cette propriété, on applique toutes les règles d'instanciation qui transforment le n-uplet de la question en un n-uplet possédant la même arité que la propriété solution.

4.3.3 Exemple

Soit la question :
Hypothèses : (AB) // (CD) et ...?
Théorème : Un quadrilatère qui a ses côtés opposés parallèles deux à deux est un parallélogramme
Conclusion : ABCD est un parallélogramme.

Cette question est fournie au système sous la forme :

parallèle, ...? → parallélogramme	(squelette de théorème incomplet)
(ABCD)	(quadruplet de points)
(AB)//(CD), AB = CD, (AD)//(BC), AD = BC, ...	(description de la figure)

Les analogies du premier type produisent les réponses suivantes :

- L'analogie avec une propriété de la même question retourne "(AB) // (CD)" et "ABCD est un parallélogramme"

- L'analogie avec une propriété de même statut retourne la ou les hypothèse(s) de la question précédente et de la question suivante.

- L'analogie avec une propriété de la figure retourne "(AB)//(CD)", "AB = CD", "(AD)//(BC)", "AD = BC", etc.

Les analogies du second type sont déterminées en deux temps.

a) Détermination de la propriété manquante

Le squelette de théorème incomplet est comparé aux théorèmes de la base. L'appariement réussit dans deux cas :

- parallèle, parallèle → parallélogramme (Le théorème de la question)

- parallèle, égale → parallélogramme ("Un quadrilatère qui a deux côtés opposés parallèles et égaux est un parallélogramme")

Deux propriétés sont retenues comme solutions : "parallèle" dans le premier cas, "égale" dans le second.

b) Instanciation des solutions

Le n-uplet de la question (ABCD) est d'arité 4. La première solution, (parallèle (x,y)), est d'arité 2. Toutes les règles de type $4 \rightarrow 2 \times 2$ ou bien $4 \rightarrow 2$ sont donc applicables. Les solutions finales produites sont "(AB) // (CD)", "(AD) // (BC)", "(A) // (C)", etc.

La seconde solution (égale(x,y)) est d'arité 2 également. De la même façon, les règles de type $4 \rightarrow 2 \times 2$ ou bien $4 \rightarrow 2$ s'appliquent. Les solutions finales produites sont "AB = CD", "AD = BC", "A = C", etc.

On note que la réponse attendue (AD) // (BC) figure parmi les solutions produites.

4.3.4 Application au diagnostic

Le simulateur a fonctionné sur les questions du test et la quasi-totalité des erreurs observées sur les copies a été reproduite. La validation d'un tel programme reste difficile, faute de critères sûrs. Un certain nombre de choix (par exemple la séparation entre choix de la propriété et choix de l'instanciation) ont été faits de manière empirique et se sont révélés efficaces *a posteriori*.

Nous n'attendons pas du simulateur qu'il reproduise le fonctionnement cognitif des élèves, mais seulement qu'il fournisse globalement les mêmes réponses. L'objectif ultime du diagnostic est de permettre la remédiation : seuls sont utiles les aspects du comportement de l'élève qui pourront être pris en compte pour choisir l'action tutorielle appropriée.

Le problème peut se définir ainsi : étant donné une question Q et une réponse R fournie par un élève, trouver la ou les analogies employées ainsi que les contraintes non satisfaites. Si aucune analogie ne peut être trouvée, il reste cependant possible de déterminer les contraintes non satisfaites.

Etant donné la taille du domaine traité et le faible nombre de réponses produites, le diagnostiqueur utilise simplement la méthode "générer et tester" : le simulateur produit tous les couples (réponse, explication) possibles, et seuls ceux correspondant à la réponse recherchée sont retenus.

Exemple Soit la question :
Hypothèses : ABCD est un parallélogramme
Théorème : Un parallélogramme a ses côtés opposés parallèles deux à deux.
Conclusion : ...?

La réponse "ABCD est un parallélogramme" reçoit deux explications : recopie de l'hypothèse ou bien lecture de la figure. Les contraintes non respectées sont structurelles (hypothèse et conclusion différentes) et contextuelles (prise en compte du théorème).

La réponse "(A)//(C) et (B)//(D)" reçoit une explication : application du théorème attendu mais avec une règle d'instanciation différente. Les contraintes non respectées sont sémantiques (propriété vraie sur la figure) et de typage ("parallèle" s'applique à des droites).

La réponse "AB = CD" reçoit deux explications : application du théorème "Un parallélogramme a ses côtés opposés **égaux** deux à deux" ou bien lecture de la figure. La contrainte non respectée est contextuelle (prise en compte du théorème).

Chapitre 5

La classe

Classiquement, on distingue deux approches en EIAO : l'une orientée vers la théorie, dans laquelle on cherche à faire progresser la connaissance dans les disciplines connexes telles que l'intelligence artificielle ou la psychologie cognitive, l'autre orientée vers les applications, dans laquelle on cherche à réaliser des logiciels utilisables dans des situations pédagogiques concrètes [Bruillard 91].

Tout au long de notre travail, nous avons tenté de concilier ces deux aspects dans le projet Mentoniezh. Ce chapitre est plus spécialement consacré à la seconde approche, l'usage du tuteur dans la classe.

Afin de faciliter l'intégration de l'outil dans la progression du cours, la possibilité a été donnée à l'enseignant de paramétrer le tuteur. Plusieurs maquettes du démonstrateur et du tuteur ont été développées au fur et à mesure de l'avancement du projet. Elles ont donné lieu à des expérimentations en collège, dont les résultats ont été analysés. Les prototypes les plus aboutis ont été diffusés, et des enseignants ont ainsi pu concevoir des séquences pédagogiques intégrant le logiciel. En particulier, des professeurs stagiaires de l'IUFM ont utilisé le tuteur dans le cadre de leur mémoire professionnel. Enfin, nous avons étudié la possibilité de faire coopérer deux tuteurs de géométrie, l'un portant sur la figure (TALC) et l'autre sur la preuve (Mentoniezh).

5.1 Consignes pédagogiques et règles implicites

Le tuteur Mentoniezh peut être utilisé par un élève seul, mais aussi dans les classe, en présence de l'enseignant. Dans cette seconde situation, il doit pouvoir s'adapter aux objectifs particuliers du professeur qui varient selon le niveau des élèves et les connaissances qu'ils ont acquises. Cette adaptation se fait par paramétrage : le professeur fixe un ensemble de « consignes pédagogiques » propres à un élève et un exercice donné, qui déterminent le comportement du tuteur pour cette session. Les consignes portent sur les points suivants :

- le nombre minimum d'observations à effectuer pendant la phase de lecture. Par défaut, aucune observation n'est exigée de l'élève. Le professeur peut obliger l'élève à citer un certain nombre de propriétés lues sur la figure.

- le sens de la démonstration. Par défaut, l'élève peut commencer une démonstration par le début, la fin ou le milieu. Le professeur peut imposer d'effectuer la preuve en partant des données, selon une démarche linéaire uniquement.

- les étapes inutiles ou impossibles (cf. 2.2.2). Par défaut, elles sont acceptées. Le professeur peut décider de refuser les unes ou les autres.

- les théorèmes « interdits ». Par défaut, tous les théorèmes de la base sont accessibles. Le professeur peut décider d'en masquer certains, soit parce que l'élève ne les connaît pas encore, soit parce qu'il veut écarter certaines solutions. Ces théorèmes n'apparaîtront pas à l'interface.

- les théorèmes implicites. Ce qui différencie généralement la preuve donnée par un expert de celle donnée par un novice est le taux d'implicites. L'expert ignore des conclusions intermédiaires qui lui paraissent évidentes. Même en se limitant au cadre du système scolaire, il est impossible de définir une « preuve idéale » pour un problème donné. En effet, les enseignants n'ont pas tous les mêmes exigences quant au degré de détail qu'ils attendent de la part des élèves. De plus, l'enseignant demande des justifications précises au début de l'apprentissage de la démonstration, pour accepter de plus en plus d'implicites au fur et à mesure que les élèves progressent. Un tuteur intelligent doit donc pouvoir modifier son niveau d'explicitation afin de s'adapter à la situation. Dans Mentoniezh, les théorèmes sont explicites par défaut, mais le professeur a la possibilité d'en déclarer certains comme implicites. Le degré d'implicite peut donc varier selon la difficulté du problème, le niveau de l'élève ou le stade de l'apprentissage. La mise en œuvre de cette consigne est plus complexe que les précédentes et demande des modifications au niveau du démonstrateur et du tuteur.

Les implicites dans le démonstrateur

Les implicites étudiés ici portent sur un pas de preuve, c'est-à-dire que l'on considère le cas où une étape de la démonstration est passée sous silence. Les implicites portant sur les énoncés (omettre de répéter, en hypothèse d'un pas, la conclusion déduite au pas précédent) ne sont pas traités dans Mentoniezh, en raison du mode de fonctionnement pas à pas du tuteur.

Les règles du démonstrateur sont partagées en deux groupes : les règles implicites et les règles explicites. Un petit nombre de règles est toujours implicite : celles pour lesquelles les enseignants s'accordent pour dire qu'elles sont implicites même en début d'apprentissage (par exemple, la transitivité de l'alignement des points). Ceci est l'équivalent des implicites de Chypre. Les autres ont un statut variable. Elles sont explicites par défaut, mais le professeur peut en définir une ou plusieurs comme implicites. Lorsqu'une règle a le statut d'implicite dans un problème, tous les pas de preuve obtenus avec cette règle sont des *pas implicites*.

Dans le démonstrateur, règles implicites et explicites sont traitées de manière strictement identique, à l'exception du cas où la profondeur de recherche est bornée : la taille d'une preuve n'est pas le nombre total de pas qu'elle comporte, mais le nombre de pas explicites uniquement.

Le traitement des deux types de pas diffère au moment de la mémorisation des preuves, lorsque le démonstrateur transmet ses résultats au tuteur. En effet, les démonstrations que

manipule le tuteur ne sont composées que de pas explicites. Les preuves sont donc filtrées et débarrassées de leurs pas implicites, qui sont mémorisés à part. Ceci peut entraîner une diminution du nombre total de preuves, puisqu'il arrive que deux preuves ne diffèrent que par leurs implicites. Dans ce cas, le doublon est supprimé.

Les implicites dans le tuteur

Pour étudier les différentes manières dont l'élève peut effectuer une preuve avec des implicites dans Mentoniezh, considérons l'exemple suivant. Soient d_1 et d_2 des données, c une conclusion finale, i_1 et i_2 des conclusions intermédiaires, et la preuve :

$d_1 \Rightarrow i_1$
$i_1 \Rightarrow i_2$
$d_2, i_2 \Rightarrow c$

dans laquelle $i_1 \Rightarrow i_2$ est un pas implicite.

Dans une première version du tuteur, les implicites étaient traités seulement au niveau de l'enchaînement, puis il s'est avéré nécessaire de traiter également les implicites au niveau du pas.

Implicites au niveau de l'enchaînement On considère que l'implicite est au niveau de l'enchaînement lorsque la déduction implicite est faite entre (au sens chronologique) deux déductions explicites. Sur l'exemple, la preuve comportant un tel implicite est :

$d_1 \Rightarrow i_1$
$d_2, i_2 \Rightarrow c$

Pour reconnaître et accepter ce type de preuve, le mécanisme de mise à jour de l'état de la démonstration a été enrichi. Après chaque déduction effectuée par l'élève, on parcourt la liste des pas implicites. Tous ceux dont les hypothèses font partie des faits démontrés par l'élève sont déclenchés, c'est-à-dire que leur conclusion devient elle-même un fait démontré. Ce parcours est répété jusqu'à ce qu'il n'y ait plus aucun implicite applicable. On autorise ainsi plusieurs implicites consécutifs.

Dans la pratique, le nombre de règles implicites est suffisamment faible pour que cette phase de déduction ne perturbe pas les temps de réponse du tuteur.

Implicites au niveau du pas Au cours des expérimentations du tuteur en classe, il s'est avéré que les élèves cherchaient parfois à utiliser une formulation dans laquelle l'implicite est présent à l'intérieur d'un pas et non entre deux pas. Sur l'exemple, on peut proposer deux preuves comportant ce type d'implicite :

$d_1 \Rightarrow i_2$
$d_2, i_2 \Rightarrow c$

et aussi :

$d_1 \Rightarrow i_1$
$d_2, i_1 \Rightarrow c$

Pour des raisons évidentes de cohérence, cette formulation doit être admise par le tuteur au même titre que la formulation précédente. Ceci requiert de modifier non plus le mécanisme de mise à jour de l'état de la démonstration, mais celui de la validation du pas proposé par l'élève. Jusque là, le pas de l'élève était accepté *si et seulement si* le démonstrateur avait

produit un pas identique. La notion de validité a donc été élargie à d'autres cas. Reconnaître un implicite au niveau du pas revient à détecter une sous-preuve à l'intérieur du pas. Autrement dit, on recherche un ensemble de pas tel que, si (h_1,...h_n $\Rightarrow$ c) désigne le pas de l'élève :

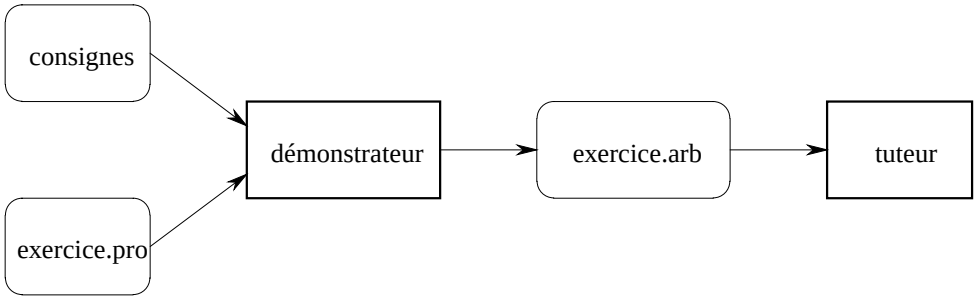
- exactement un de ces pas est explicite,
- tous les autres sont implicites,
- cet ensemble forme une preuve stricte de c à partir de h_1,...h_n, c'est-à-dire que toutes les h_i et toutes les conclusions intermédiaires sont utilisés dans la preuve (on se limite aux preuves de profondeur deux).

Si une telle sous-preuve existe, alors la proposition de l'élève est jugée correcte. Ici encore, le nombre de pas implicites est assez faible pour que la recherche systématique des sous-preuves ne dégrade pas les temps de réponse.

5.2 Réalisation et développements

Le démonstrateur et le tuteur ont donné lieu à des développements successifs. Les premières maquettes ont été réalisées en Prolog II sur station Sun. Une version a ensuite été portée sur compatible PC, afin de permettre des expérimentations en collège. Enfin, en 1996, le tuteur a été entièrement réécrit en Turbo Pascal 6.0 sur PC, afin de bénéficier d'une interface plus agréable.

Le démonstrateur produit, à partir d'un fichier *.pro* contenant l'énoncé d'un problème et d'un fichier de consignes pédagogiques, un fichier *.arb* contenant l'espace de recherche structuré de ce problème, qui est utilisé par le tuteur. Pour la version en Pascal, un traducteur a été écrit afin de produire, à partir d'un fichier *.arb* respectant la syntaxe Prolog, un fichier texte directement exploitable par le programme Pascal.



Le langage Prolog permet une traduction assez directe des énoncés HDL : les objets géométriques sont représentés par des identificateurs de constantes et les propriétés géométriques par des termes. Les deux listes de faits, Faits-étudiés et Faits-à-étudier sont gérées en pile en utilisant le mécanisme des assertions (*assert* et *suppress*) de Prolog II.

Voici par exemple le fichier décrivant l'exercice présenté en 2.1.3.

```
points(A.B.I.J.P.nil) -> ;
```

```
droites(nil) -> ;
cercles(C.nil) -> ;
convexe(A,B,I,J) -> ;
droite(dr1,A.I.P.nil) -> ;
droite(dr2,B.J.P.nil) -> ;
enonce(1,"Soit C un cercle de diamètre [AB]. P est un") -> ;
enonce(2,"point quelconque de C. Les points I et J sont") -> ;
enonce(3,"tels que P est le milieu de [AI] et de [BJ].") -> ;
enonce(4,"Démontre que ABIJ est un losange.") -> ;
fait-a-etudier(1,diametre(A,B,C)) -> ;
fait-a-etudier(2,app-cercle(P,C)) -> ;
fait-a-etudier(3,milieu(P,A,I)) -> ;
fait-a-etudier(4,milieu(P,B,J)) -> ;
but(losange(A,B,I,J)) -> ;
```

Chaque théorème ou définition est exprimé par un paquet de règles Prolog, de telle manière qu'un théorème ayant *n* prémisses sera représenté par *n* règles au minimum (les symétries pouvant entraîner une duplication de règles). Les théorèmes ne sont donc pas exprimés sous forme déclarative. Ceci permet de traiter au cas par cas les symétries et d'obtenir une efficacité satisfaisante du démonstrateur. En contrepartie, cela interdit à l'enseignant d'ajouter des théorèmes car l'extension de la base nécessite de rédiger de nouvelles règles Prolog.

5.3 Validation et expérimentations en classe

A partir de 1991, le tuteur a été expérimenté chaque année, à raison d'une ou deux séries d'évaluations comprenant de quatre à six séances d'une heure de travail. Ces expérimentations se sont déroulées en classe de quatrième dans deux collèges de la région rennaise, avec la collaboration du groupe rennais du GR Didactique et en particulier de deux professeurs de mathématiques, Geneviève Mouraud et Christian Boulard. Les classes concernées étaient de niveaux très variables.

Les expérimentations se sont déroulées selon le scénario suivant. Le choix des problèmes et des consignes est fait au préalable par l'enseignant. Les problèmes retenus comportent en général trois ou quatre pas de preuve. Les élèves travaillent par binômes ou seuls. À chaque séance, ils reçoivent l'énoncé d'un problème. Ils commencent par tracer la figure sur papier, puis résolvent le problème sur ordinateur. Le professeur circule dans la classe et répond aux questions ou aide les élèves en difficulté. À la fin d'une série d'expérimentations, les élèves remplissent un questionnaire à propos du logiciel.

Toutes les traces d'interaction élève/tuteur étaient enregistrées, et plusieurs enregistrements audio ont été réalisés. La majorité des traces a été étudiée *a posteriori* par les professeurs [Py et El Gass 92]. Ce travail d'analyse a conduit les enseignants à suggérer des modifications du tuteur. C'est à leur demande que les messages d'erreur et le diagnostic ont été modifiés, et que la notion d'implicite a été introduite.

Par ailleurs, une analyse didactique *a priori* de Mentoniezh a été réalisée par Vanda Luengo [Luengo 93], mais il n'y a pas eu d'étude didactique portant sur l'utilisation en

classe du tuteur. On peut néanmoins s'essayer à dégager quelques conclusions à partir de ces expériences.

Par rapport à une situation papier-crayon, la situation avec Mentoniezh présente un certain nombre de contraintes nouvelles : contraintes de formulation des énoncés, de validité des assertions, d'organisation de la preuve. Ces contraintes s'avèrent productives, du point de vue de l'apprentissage, dans plusieurs cas. Par exemple, l'obligation d'extraire, du texte de l'énoncé, chaque propriété, et de l'étiqueter *hypothèse* ou *conclusion* amène rapidement (après deux ou trois séances) les élèves à une meilleure perception du statut des faits. En phase de démonstration, le fait que le tuteur exige, pour juger de la validité d'un pas, que soient fournis à la fois les hypothèses, le théorème et la conclusion fait prendre conscience aux élèves de la structure ternaire du pas. On constate que le message associé à cette erreur, fréquent pendant les premières séances, est de moins en moins déclenché. De même, en début d'apprentissage, les élèves décalquent les techniques apprises sur la démonstration à un pas et cherchent à résoudre le problème en une seule étape, qui agglomère plusieurs théorèmes de manière plus ou moins malhabile. Le tuteur les oblige à décomposer la preuve en autant d'étapes qu'il y a de théorèmes à utiliser, facilitant ainsi la compréhension de la structure de la preuve.

A l'opposé, ces contraintes peuvent aussi constituer des obstacles qui gênent l'expression de la preuve par l'élève. La liste des propriétés prédéfinies dans le tuteur (qui correspond aux termes HDL) ne permet pas de rendre compte de la variété des formulations possibles en français. Ainsi, « le point M appartient à la droite (AB) » se traduira par « A, M, B sont alignés ». De plus, certains énoncés ne se traduisent pas : par exemple, « ABC est un triangle » ne constitue pas une propriété, pour le tuteur, puisque trois points non alignés forment toujours un triangle. Cette distance entre langage mathématique et langage du tuteur provoque des effets indésirables : lorsqu'il ne réussit pas à exprimer la propriété visée, l'élève abandonne ou propose une propriété incorrecte en remplacement. Si des solutions locales peuvent être trouvées à ces problèmes, en revanche la question générale de l'écart entre représentations informatiques et papier-crayon reste ouverte. On touche ici au phénomène de la « transposition informatique », processus d'adaptation des savoirs aux contraintes de modélisation et d'implantation informatiques [Balacheff 94].

Il arrive aussi que l'élève s'adapte aux contraintes en les contournant : il trouve une stratégie locale qui lui permet de résoudre les problèmes avec l'aide du tuteur, mais qui n'est ni généralisable, ni transposable en papier-crayon. Déstabiliser ce type de stratégie suppose que le mode d'interaction avec le tuteur évolue. L'enseignant peut agir sur certains leviers, en modifiant les consignes ou en choisissant des problèmes pour lesquels la stratégie de l'élève échouerait. Cependant, pour prendre en compte ce problème de manière globale, il serait nécessaire de définir plusieurs formes d'interaction élève/tuteur et de pouvoir adopter l'une ou l'autre en fonction des objectifs visés, ou encore de permettre à plusieurs environnements, intégrant des modèles d'interaction différents, de coopérer entre eux.

Les expérimentations concernaient des classes de niveaux très hétérogènes, ce qui nous a permis d'effectuer des comparaisons. Il semble que le comportement des élèves face au tuteur dépende fortement de la connaissance préalable qu'ils ont de la démonstration et du sens qu'ils lui attribuent. On distinguerait ainsi trois types d'élèves :

- ceux qui maîtrisent déjà la démonstration. Ils font une utilisation rationnelle du tuteur

et apprennent rapidement à exploiter ses possibilités. Par exemple, ils consultent la base de théorèmes à bon escient et anticipent sur le déroulement de la démonstration. Ils ont une idée assez précise de la preuve qu'ils vont réaliser avant de commencer leur travail et ne font que des erreurs mineures. Lorsqu'ils sont en désaccord avec le tuteur, ils le contestent et demandent l'arbitrage du professeur.

- ceux qui maîtrisent partiellement la démonstration. Ils manipulent correctement le tuteur sans exploiter toutes ses possibilités. Ils ont ébauché mentalement la preuve qu'ils vont réaliser. L'interaction avec le tuteur leur permet de corriger et de compléter cette ébauche.

- ceux qui n'ont pas compris le principe de la démonstration. Ils rencontrent des problèmes de manipulation du tuteur à tous les stades, même pour les commandes les plus simples. Ils commencent sans avoir aucune idée de la preuve qu'ils vont réaliser, et tâtonnent en permanence. Ils finissent par aboutir à force d'essais successifs et de réussites locales. Lorsqu'ils sont en désaccord avec le tuteur, ils adoptent une attitude docile et partent dans une autre direction plutôt que de contester.

L'acquisition du sens profond de la démonstration et de la technique de la démonstration sont deux aspects fortement liés [Arsac 88]. Il semblerait de plus que la maîtrise et l'utilisation rationnelle d'un EIAO de preuve soit liée à la compréhension du sens de la démonstration. En effet, la logique d'un tuteur de preuve s'inspire de la logique de la démonstration : un élève qui a assimilé cette dernière se retrouve en terrain familier. Un tuteur tel que Mentoniezh peut donc apporter une aide à la maîtrise technique de la preuve, mais la construction de situations qui permettent aux élèves d'acquérir le sens de la preuve reste le rôle du professeur.

5.4 Utilisation dans l'enseignement

La version écrite en Prolog a été diffusée sur disquettes à une centaine d'exemplaires, en France et dans des pays francophones. La version écrite en Turbo-Pascal (appelée *Tigre*) est téléchargeable sur le site web de l'Irisa [1].

Le tuteur Mentoniezh a également été utilisé par des professeurs stagiaires au cours de leur travail de mémoire professionnel [Denais et Gesret 99] [Feulvarch et Mimeau 98]. Michel Denais et Anne Gesret ont cherché à combiner différents outils et méthodes (papier-crayon, travail collectif, logiciel) pour l'acquisition de la démonstration. Ils ont plus particulièrement étudié la phase de recherche d'une solution, et ont élaboré une séquence pédagogique destinée à amener les élèves à s'approprier d'une part le schéma déductif si-alors et d'autre part la méthode de recherche descendante. Cette séquence combine une fiche d'activités écrites, un travail de groupe par oral avec un rétroprojecteur et une séance sur ordinateur avec Mentoniezh. En conclusion, ils observent que le travail sur la phase de recherche descendante a permis d'éclaircir le rôle des données et de la conclusion. L'utilisation de différents supports a fait appel aux capacités tant écrites qu'orales des élèves, et l'outil informatique a permis de développer leur autonomie. La séquence a également été profitable aux professeurs stagiaires, car elle leur a permis d'observer finement les procédures mises en place chez les élèves et de remettre en cause certaines pratiques de leur enseignement.

1. http://www.irisa.fr/aida/Dominique.Py/Tigre/

Enfin, Mentoniezh est utilisé par des formateurs d'IUFM dans le cadre de la formation des enseignants stagiaires [Bazin et al. 99]. Les auteurs considèrent que l'emploi du logiciel permet de révéler les termes du contrat didactique et de mettre en évidence le poids des implicites dans l'enseignement.

5.5 Interopération entre EIAOs

La recherche en EIAO a donné lieu à de multiples applications, mais chacune ne traite qu'un domaine très spécialisé, étant donné la complexité de leur réalisation. C'est pourquoi la nécessité de faire coopérer et d'intégrer ces logiciels au sein de plate-formes est actuellement ressentie.

Avec Cyrille Desmoulins et Marilyne Macrelle, nous avons étudié la possibilité de faire interopérer deux tuteurs de géométrie, l'un consacré à la figure (TALC) et l'autre à la démonstration (Mentoniezh). Cette coopération est facilitée du fait que les deux logiciels sont issus du même projet (cf. [Trilling 96]). Ils utilisent, pour décrire les figures géométriques, des représentations assez voisines et sont programmés en Prolog II. Mentoniezh se base sur le langage HDL et TALC sur le langage CDL : ce sont des langages déclaratifs, dont les phrases sont des prédicats portant sur des identificateurs ou des termes (en CDL). La granularité de CDL est plus faible que celle de HDL, si bien que toute propriété exprimée en HDL peut se traduire en CDL, à l'exception de la convexité du quadrilatère (qui fait intervenir la négation).

La traduction d'un langage de description à un autre s'effectue au moyen de macro-définitions [Macrelle et Desmoulins 98]. Pour chaque phrase HDL sont définies une ou plusieurs macro-CDL. Par exemple :

terme HDL $\rightarrow$ conjonction de termes CDL

$parallele(d1,d2) \rightarrow droite(d1),droite(d2),d1//d2$

$alignes(a,b,c) \rightarrow a \in (b,c)$
$alignes(a,b,c) \rightarrow (ab) = (ac)$
$alignes(a,b,c) \rightarrow (ab)//(ac)$

Lorsque plusieurs définitions coexistent pour un même terme, c'est à l'enseignant de déterminer celle qui doit être retenue, en fonction de l'énoncé à traiter. La traduction est effectuée par un interpréteur. Celui-ci prend en entrée d'une part la liste des macro-définitions pour Mentoniezh et d'autre part un énoncé HDL. Il produit un énoncé CDL en procédant par réécriture à partir des formules de l'énoncé HDL.

Ce travail constitue une première étape vers une plate-forme permettant l'interopérabilité, au niveau des connaissances, entre deux tuteurs de géométrie.

Chapitre 6

Conclusion

Le travail autour d'une application unique durant plusieurs années, loin de constituer une entrave, m'a au contraire permis d'explorer de manière approfondie les facettes de l'EIAO et leurs relations complexes. Tout au long du projet, j'ai tenté de maintenir un équilibre entre la recherche, le développement et les expérimentations de terrain. De nombreuses pistes restent à prospecter, tant sur les aspects théoriques du domaine que sur la question de l'usage des environnements d'apprentissage.

Le modèle de l'expert, ou modèle du domaine, est le thème qui a donné lieu au plus grand nombre de recherches en EIAO. Il n'est pas surprenant que ce soit aussi le thème le plus souvent étudié au sein du projet Mentoniezh. Si la géométrie élémentaire enseignée au collège peut paraître *a priori* un domaine simple, elle n'en recèle pas moins d'intéressants problèmes de représentation et de modélisation, comme en témoignent les travaux qui s'y sont attaqués sous des angles variés. L'originalité du travail mené dans Mentoniezh tient à la recherche d'un ensemble (si possible exhaustif) de preuves, là où les démonstrateurs usuels établissent une seule preuve. Nous avons montré la faisabilité et l'intérêt de cette approche sur le domaine considéré. L'emploi de la saturation, en produisant les propriétés valides, pallie l'absence de figure et permet l'élaboration d'un « expert explorateur » ainsi que l'introduction de nouveaux objets. L'énumération des preuves procure les connaissances nécessaires à un tuteur de type « expert encadreur ». En contrepartie, cette approche génère des difficultés spécifiques liées à la représentation des droites et des preuves. Des solutions ont été proposées pour gérer dynamiquement les droites tout en conservant une complexité raisonnable.De plus, des critères ont été définis pour examiner les démonstrations de manière plus qualitative (élégance ou lisibilité) que le simple critère du nombre de pas de preuve. Il serait intéressant de chercher une représentation **structurée** d'un ensemble de preuves, par exemple sous forme d'un graphe matérialisant les similitudes et les différences entre éléments, ou encore d'avoir une représentation des preuves **à granularité variable** qui faciliterait la mise en évidence de familles de preuves.

La définition des scénarios d'interaction dans Mentoniezh découle du découpage en tâches de l'activité de démonstration. Par rapport à d'autres tuteurs qui se focalisent sur la

démonstration, la séquence des activités dans Mentoniezh mène de l'énoncé du problème à la démonstration rédigée. La tâche initiale de lecture de l'énoncé et la tâche finale de rédaction sont relativement simples, tandis que la tâche de démonstration laisse à l'élève la possibilité de développer et d'expérimenter des stratégies variées, et que la tâche d'exploration est assimilable à un parcours d'espace de recherche, guidé et encadré par le tuteur. Cette approche de la recherche heuristique d'une solution, conçue comme un dialogue dirigé par le tuteur, n'a pas d'équivalent dans les autres systèmes. Seul le logiciel DEFI comporte une phase d'exploration de ce type, mais celle-ci fonctionne de manière très rigide en raison d'une base de théorèmes limitée et de l'absence de résolveur.

Le tuteur enchaîne les tâches de manière séquentielle, alors qu'en mode papier-crayon, l'élève est souvent amené à faire des allers-retours entre les différentes étapes. Une extension possible consisterait à permettre à l'élève de naviguer entre les différents modules, sans lui imposer de valider une étape avant d'aborder la suivante.

La modélisation de l'élève est abordée sous deux angles, local et global. Le diagnostic des erreurs est effectué au niveau du pas de démonstration. Ce diagnostic possède une meilleure expressivité que les classiques catalogues de règles erronées (basés sur l'appariement de la réponse de l'élève avec un patron de réponse incorrecte) dans la mesure où il s'attache à reconnaître le pas de preuve visé par l'élève, en s'aidant du contexte. Cette approche oblige à une certaine décompilation des règles de diagnostic, ce qui les rend plus lisibles et mieux compréhensibles par un utilisateur du système. La détection de la stratégie mise en œuvre par l'élève s'appuie sur une méthode de reconnaissance de plan qui a été adaptée à la situation de résolution de problème. Cette méthode tire sa robustesse de l'emploi de plans objectifs, qui sont ici les solutions. La reconnaissance de plan est une technique de modélisation de l'élève plus puissante que le simple *model-tracing* implanté dans Geometry Tutor, qui ne porte que sur le pas de preuve courant : elle permet une analyse globale et fournit un modèle prédictif des actions de l'élève.

Dans les deux cas, il s'agit de modélisation comportementale : aucune hypothèse n'est faite sur les connaissances détenues par l'élève. Parallèlement, nous avons exploré une approche de type "modélisation cognitive" en restreignant pour cela le domaine à la démonstration en un pas. Ce travail constitue une première étape vers l'élaboration d'un modèle *exécutable* de l'élève, domaine qui n'a pas été abordé, jusqu'à présent, pour les tâches de résolution de problèmes telles que la démonstration. En effet, la modélisation des connaissances de l'élève sur cette famille de tâches reste un problème ouvert. L'articulation entre les deux niveaux de modélisation, preuve et pas, demande tout particulièrement à être éclaircie.

Dans le tuteur Mentoniezh, les résultats obtenus servent actuellement à sélectionner le message d'erreur le plus pertinent. Ils pourraient être exploités à d'autres fins, par exemple pour élaborer des explications ou des conseils.

L'expérimentation en classe et la diffusion du logiciel auprès des enseignants constitue l'aboutissement du processus de conception d'un EIAO. Les prototypes successivement réalisés ne sont pas assimilables à des produits finis, cependant ils ont permis de valider les hypothèses qui fondent le travail de recherche. La principale lacune pointée par les utilisateurs est que le logiciel ne permet pas de manipuler la figure géométrique. L'intégration,

dans une plateforme commune, de logiciels dédiés à des tâches différentes permettrait de surmonter cet inconvénient.

Si les EIAO sont conçus dans le but d'un usage par l'élève, dans la pratique, diverses utilisations d'un EIAO directement par l'enseignant sont possibles : pour la formation des enseignants, pour l'analyse *a priori* d'un problème, pour le suivi des élèves. Ces usages n'étaient pas prévus explicitement par le concepteur et résultent souvent d'un détournement de l'outil. Une piste intéressante serait de les prendre en compte *a posteriori* afin de développer des fonctionnalités plus directement adaptées à l'usage réel que font les enseignants de l'EIAO, telles que le recensement des solutions d'un exercice ou la production d'une synthèse de l'activité de l'élève au cours de la session.

Bibliographie

[Anderson et al. 85] J.R. Anderson, C. F. Boyle et G. Yost, « The geometry tutor », *Proceedings of Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, 1985, p. 1-7.

[Arsac 88] G. Arsac, « Les recherches actuelles sur l'apprentissage de la démonstration et les phénomènes de validation en France », *Recherches en didactique des mathématiques*, vol. 9, n° 3, La Pensée Sauvage, Grenoble, 1988, p. 247-280.

[Artigue 92] M. Artigue, « Mathématiques : les leçons d'une crise », *Science et Vie*, n° 180, 1992, p. 46-59.

[Artigue 97] M. Artigue, « Une approche didactique de l'intégration des EIAO à l'enseignement », *Environnements Interactifs d'Apprentissage avec Ordinateur*, D. Guin, J.-F. Nicaud et D. Py (eds), Eyrolles, Paris, 1995, p. 17-28.

[Bachelard 38] G. Bachelard, « *La formation de l'esprit scientifique* », Vrin, Paris, 1975.

[Balacheff 94] N. Balacheff, « *La transposition informatique. Note sur un nouveau problème pour la didactique* », *Vingt ans de didactique des mathématiques en France*, La Pensée Sauvage, Grenoble, 1994, p. 364-370.

[Barz et Holland 89] W. Barz et G. Holland, « Intelligent Tutoring Systems for training in geometrical proof and construction problems », *Learning and Instruction: European research in an international context*, H. Mandl, E. De Corte et S.N. Bennett (eds), Pergamon, Oxford, 1989.

[Baulac et Giorgiutti 91] Y. Baulac et I. Giorgiutti, « Interaction micromonde/tuteur en géométrie », *Actes des deuxièmes journées EIAO de Cachan*, éditions de l'ENS de Cachan, 1991, p. 11-18.

[Bazin 93] J.-M. Bazin, « GÉOMUS *: un résolveur de problèmes de géométrie qui mobilise ses connaissances en fonction du problème posé* », thèse de l'université de Paris VI, 1993.

[Bazin et al. 99] J.-M. Bazin, M. Mézière et J. Vincent « Un exemple d'utilisation d'un EIAO pour former des enseignants », *Troisième congrès international d'actualité de la recherche en éducation et en formation*, Bordeaux, 1999.

[Bernat 94] Ph. Bernat, « *Conception et réalisation d'un environnement interactif d'aide à la résolution de problèmes.* CHYPRE *: un exemple pour la démonstration en géométrie* », thèse de l'université de Nancy, 1994

[Bouhineau 97] D. Bouhineau, « Construction automatique de figures géométriques et programmation logique », thèse de l'université Joseph Fourier, Grenoble, 1997.

[Brousseau 83] G. Brousseau, « Les obstacles épistémologiques et les problèmes en mathématiques », *Recherches en didactiques des mathématiques*, vol. 4, n° 2, La Pensée Sauvage, Grenoble, 1983, p. 165-198.

[Bruillard 91] E. Bruillard, « *Mathématiques et enseignement intelligemment assisté par ordinateur* », thèse de l'université du Maine, 1991.

[Bruillard 97] E. Bruillard, « *Les machines à enseigner* », Hermès, Paris, 1997.

[Burton 82] R. Burton, « Diagnosing bugs in a simple procedural skill », *Intelligent Tutoring Systems*. D. Sleeman et J.S. Brown (eds), Academic Press, New-York, 1982, p. 157-184.

[Carpentier 98] F.-G. Carpentier, « *Modélisation des connaissances et de la démonstration pour l'EIAO de la géométrie* », thèse de l'université de Rennes I, 1998.

[Coelho et Pereira 79] H. Coelho et L.M. Pereira, « GEOM *: a Prolog geometry theorem prover* », rapport interne n° 525, Laboratoire National d'Ingéniérie Civile, Lisbonne, 1989.

[Denais et Gesret 99] M. Denais et A. Gesret, « *Complémentarité de différents outils dans l'apprentissage de la démonstration* », mémoire professionnel, IUFM de Bretagne, Rennes, 1999.

[Desmoulins 94] C. Desmoulins, « *Etude et réalisation d'un système tuteur pour la construction de figures géométriques* », thèse de l'université Joseph Fourier, Grenoble, 1994.

[Delozanne 92] E. Delozanne, « *Explications en EIAO : études à partir d'*ELISE*, un logiciel pour entraîner à une méthode de calcul des primitives* », thèse de l'université du Maine, Le Mans, 1992.

[Duval et Egret 89] R. Duval et M.-A. Egret, « L'organisation déductive du discours », *Annales de Didactique et de Sciences Cognitives*, vol. 2, IREM de Strasbourg, 1989, p. 25-40.

[Feulvarch et Mimeau 98] S. Feulvarch et O. Mimeau, « Quelques activités de remédiation à la démonstration en géométrie à l'aide de l'outil informatique », mémoire professionnel, IUFM de Poitiers, 1998.

[Gaud et Guichard 84] D. Gaud et J.-P. Guichard, « Apprentissage de la démonstration », *Petit x* n° 4, IREM de Grenoble, 1984, p. 5-25.

[Gelernter 63] H. Gelernter, « Realization of a geometry theorem-proving machine », *Computers and Thought*, Feigenbaum et Feldman (eds), 1963, p. 153-163.

[Genesereth 82] M. Genesereth, « The role of plans in intelligent teaching systems », *Intelligent Tutoring Systems*, D.Sleeman et J.S.Brown (eds), Academic Press, New York, 1982, p. 137-156.

[Goldstein 82] I. P. Goldstein, « The genetic graph : a representation for the evaluation of procedural knowledge », *Intelligent Tutoring Systems*, D. Sleeman et J.S. Brown (eds), Academic Press, New-York, 1982, p. 51-78.

[Guin 89] D. Guin, « Réflexions sur les logiciels d'aide à la démonstration en géométrie », *Annales de Didactique et de Sciences Cognitives*, vol. 2, IREM de Strasbourg, 1989, p. 89-109.

[Guin et al. 91] D. Guin, M.-A. Egret, B. Koch, G. Kuntz, G. Métivier, N. Vogel, « Modélisation de la démonstration géométrique dans Geometry Tutor », *Annales de Didactique et de Sciences Cognitives*, vol. 3, IREM de Strasbourg, 1991, p. 1-20.

[Haton et al. 91] J. P. Haton, N. Bouzid, F. Charpillet, M.-C. Haton, B. Laasri, H. Laasri, P. Marquis, T. Mondot et A. Napoli, « *Le raisonnement en intelligence artificielle* », InterEditions, Paris, 1991.

[Kass 87] R. Kass, « *The role of user modelling in I.T.S.* », rapport de recherche, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, 1987.

[Kautz 87] H.A. Kautz, « *A formal theory of plan recognition* », Ph. D. Thesis, Department of Computer Science, University of Rochester, New Jersey, 1987.

[Le Pape 97] G. Le Pape, « *Démonstration automatique en géométrie élémentaire* », rapport de stage de DEA, Irisa, Rennes, 1997.

[Luengo 93] V. Luengo, « *Contraintes informatiques et apprentissage de la démonstration, à propos de trois logiciels* », mémoire de DEA de didactique des disciplines scientifiques, université Joseph Fourier, Grenoble, 1993.

[Luengo 97] V. Luengo, « CABRI-EUCLIDE, *un micromonde de preuve intégrant la réfutation. Principes didactiques et informatiques, réalisation* », thèse de l'université Joseph Fourier, Grenoble, 1997.

[Nevins 75] A. J. Nevins , « Plane geometry theorem-proving using forward chaining », *Artificial Intelligence*, n° 6, 1975, p. 1-23.

[Nicolas 89] P. Nicolas, « *Construction et vérification de figures géométriques dans le système Mentoniezh* », thèse de l'université de Rennes I, 1989.

[Noël et Py 93] C. Noël et D. Py, « Diagnostic des erreurs de l'élève basé sur le contexte », *Environnements Interactifs d'Apprentissage avec Ordinateur*, M. Baron, R. Gras et J.-F. Nicaud (eds), Eyrolles, Paris, 1993, p. 229-240.

[Laborde 95] J.-M. Laborde, « Des connaissances abstraites aux réalités artificielles, le concept de micromonde Cabri », *Environnements Interactifs d'Apprentissage avec Ordinateur*, D. Guin, J.-F. Nicaud et D. Py (eds), Eyrolles, Paris, 1995, p. 29-42.

[Macrelle et Desmoulins 98] M. Macrelle et C. Desmoulins, « Macro-definitions, a basic component for interoperability between ILEs at the knowledge level : application to geometry », *International Conference on Intelligent Tutoring Systems*, B. P. Goettl, H. M. Halff, C. L. Redfield et V. J. Shute (eds), Lecture Notes in Computer Science n° 1452, Springer, 1998.

[Ohlsson et Langley 88] S. Ohlsson et P. Langley, « Psychological evaluation of path hypotheses in cognitive diagnosis », *Learning issues for Intelligent Tutoring Systems*, H. Mandl et A. Lesgold (eds), Springer-Verlag, 1988, p. 42-62.

[Pintado 94] M. Pintado , « *Apprentissage et démonstration automatique de théorèmes* », thèse de l'université de Paris VI, 1994.

[Py 90] D. Py, « *Reconnaissance de plan pour l'aide à la démonstration dans un Tuteur Intelligent de la géométrie* », thèse de l'université de Rennes I, 1990.

[Py et Nicolas 90] D. Py et P. Nicolas, « Mentoniezh, a geometry I.T.S. for figure setting and proof drawing », *Journal of Artificial Intelligence in Education*, vol. 1, n° 3, 1990, p. 41-56.

[Py et El Gass 92] D. Py et B. El Gass, « Expérimentation du tutoriel Mentoniezh en classe de quatrième », *Petit x*, n° 30, 1992, p. 63-72.

[Py 93] D. Py « Geometry problem-solving with Mentoniezh », *Computers and Education*, Pergamon Press, vol. 20, n° 1, 1993, p. 141-146.

[Py 94] D. Py, « Les erreurs en géométrie : une étude de cas », *Actes du Séminaire Dida-Tech*, séminaire n° 160, IMAG-LSD2, Grenoble, France, 1994, p. 119-136.

[Py 95] D. Py, « Modélisation du diagnostic en EIAO », *Environnements Interactifs d'Apprentissage avec Ordinateur*. D. Guin, J.-F. Nicaud et D. Py (eds), Eyrolles, Paris, p. 185-195.

[Py 96a] D. Py, « Démonstration et implicites dans le tuteur Mentoniezh », *dixième congrès Reconnaissance des formes et Intelligence Artificielle*, AFCET-AFIA, Rennes, 1996, p. 358-366.

[Py 96b] D. Py, « Aide à la démonstration en géométrie : le projet Mentoniezh », *Sciences et techniques éducatives*, vol. 3, n° 2, Hermès, Paris, 1996, p. 227-256.

[Py 00] D. Py, « Exploration guidée dans un tuteur intelligent », *douzième congrès Reconnaissance des formes et Intelligence Artificielle*, AFRIF-AFIA, Paris, 2000, p. 141-152.

[Self 88] J. A. Self, « Student models : what use are they ? », *Artificial Intelligence tools in education*, P. Ercoli et R. Lewis (eds), IFIP, 1988, p. 73-86.

[Self 92] J. A. Self, « Cognitive diagnosis for tutoring systems », *Proceedings of Tenth European Conference on Artificial Intelligence*, B. Neumann (éd.), John Wiley and Sons Ltd, 1992, p. 699-703.

[Sleeman 82] D. H. Sleeman, « Assessing Competence in Basic Algebra », *Intelligent Tutoring Systems*, D.Sleeman et J.S.Brown (eds), Academic Press, New York, 1982, p. 185-200.

[Soloway et Johnson 85] E. Soloway et W.L. Johnson, « PROUST », *Byte*, vol. 10, n° 4, 1985.

[Sutherland 63] I. E. Sutherland, « Sketchpad : a man-machine graphical communication system », actes de *Spring Joint Computer Conference*, Detroit, Michigan, 1963, p. 329-346.

[Turquety 93] C. Turquety, « *Vers un résolveur pédagogique pour la géométrie élémentaire* », rapport de stage de DEA, Irisa, Rennes, 1993.

[Trilling 96] L. Trilling, « Rétrospective sur le projet Mentoniezh », *Sciences et techniques éducatives*, vol. 3, n° 2, Hermès, Paris, 1996, p. 157-162.

[Van Lehn 88] K. Van Lehn, « Toward a theory of impasse-driven learning », *Learning Issues for Intelligent Tutoring Systems*, H. Mandl et A. Lesgold (eds), Springer-Verlag, 1988, p. 19-41.

[Wenger 87] E. Wenger, « *Artificial Intelligence and Tutoring Systems* », Morgan Kaufmann Publishers, Los Altos, California, 1987.

Annexe A

Chronologie succincte

1984 Deux enseignants de mathématiques suivent un cours de programmation logique au CREFFIB de Rennes et suggèrent un système basé sur Prolog pour aider l'élève à résoudre un problème de géométrie. Un stagiaire de l'Irisa réalise un système d'acquisition de la figure, utilisant une tablette graphique.

1985 Le projet se structure autour de quatre phases : acquisition de la figure, transformations graphiques, exploration des propriétés, exposé de la démonstration. Pierrick Nicolas débute une thèse intitulée « Construction et vérification de figures géométriques dans le système Mentoniezh », thèse qui sera soutenue en 1989.

1986 Dominique Py débute une thèse intitulée « Reconnaissance de plan pour l'aide à la démonstration dans un Tuteur Intelligent de la géométrie », thèse qui sera soutenue en 1990.

1989 Une première version du système, comprenant un démonstrateur et un tuteur, est réalisée en Prolog II sous Unix. Les premières expérimentations sont menées avec le concours du GR didactique à Rennes (p. 43). L'interface est en mode ligne, la modélisation de l'élève se base sur la reconnaissance de plan (p. 28) et un modèle simple des erreurs (p. 31), l'enseignant peut fournir des consignes (p. 39), seule l'étape “démonstration" est implémentée (p. 24).

1990 Affichage graphique de l'arbre de preuve en cours de démonstration. Première ébauche de l'étape d'exploration (p. 23).

1991 Rénovation de l'interface, désormais en mode page. Ajout d'une étape de lecture de l'énoncé, avant la démonstration (p. 22). Début de la diffusion du logiciel auprès des enseignants (p. 45).

1992 Ajout d'une étape de rédaction, après la démonstration (p. 25). Modification en profondeur du diagnostic des erreurs pour prendre en compte le contexte (p. 32).

1993 Réalisation de l'expert explorateur (p. 18).

1994 Travail sur la démonstration à un seul pas, réalisation d'un modèle exécutable de l'élève et d'un diagnostiqueur (p. 34).

1995 Introduction des implicites dans le démonstrateur (p. 40). François-Gilles Carpentier débute une thèse intitulée « Modélisation des connaissances et de la démonstration pour l'EIAO de la géométrie », thèse qui sera soutenue en 1998 (p. 14). Collaboration avec Cyrille Desmoulins pour interfacer Talc et Mentoniezh (p. 46).

1996 Travail sur les aides dans le logiciel, en collaboration avec Jean-Claude Coulet, psychologue. Mise au point d'une grille d'analyse des démonstrations.

1997 Introduction automatique d'objets dans l'énoncé (p. 17).

1998 Développement de Tigre, version PC du tuteur Mentoniezh, et mise en ligne sur le site web de l'Irisa (p. 45).

1999 Développement de l'étape d'exploration (p. 23) et intégration dans la version Tigre.

Annexe B

Théorèmes du démonstrateur

Droites et cercles

1.1 Deux droites parallèles à une même troisième sont parallèles entre elles

1.2 Si deux droites sont parallèles, toute perpendiculaire à l'une est perpendiculaire à l'autre

1.3 Deux droites perpendiculaires à une même troisième sont parallèles entre elles

2.1 Si deux points appartiennent à un cercle de centre O, alors ils sont à égale distance de ce centre

2.2 Si un point P appartient à un cercle de centre O, et si un point P' est tel que OP = OP', alors P' appartient aussi à ce cercle

3.1 Si un point est le centre d'un cercle, alors c'est le milieu de tout diamètre de ce cercle

3.2 Si un point est le milieu du diamètre d'un cercle, alors c'est le centre de ce cercle

4.1 Si deux droites sont parallèles à une même troisième et ont un point commun, alors elles sont confondues

4.2 Si deux droites sont perpendiculaires à une même troisième et ont un point commun, alors elles sont confondues

4.3 Si deux droites sont parallèles et ont un point commun, alors elles sont confondues

5.1 Si un point est le milieu d'un segment, alors il est à égale distance des extrémités de ce segment

5.2 Si un point est le milieu d'un segment, alors la distance de ce point à une extrémité de ce segment vaut la moitié de ce segment

5.3 Si un point est situé sur un segment, et s'il est à égale distance des extrémités de ce segment, alors c'est le milieu de ce segment

6.1 Si deux distances sont égales à une même troisième, alors elles sont égales entre elles.

7.1 Si deux distances sont égales, tout distance égale au double (resp. à la moitié) de la première est égale au double (resp. à la moitié) de la seconde

8.1 Si une droite est la médiatrice d'un segment, alors elle est perpendiculaire à ce segment

8.2 Si une droite est perpendiculaire à un segment et passe par le milieu de ce segment, alors c'est la médiatrice de ce segment.

9.1 Si un point appartient à la médiatrice d'un segment, alors il est à égale distance des extrémités de ce segment

9.2 Si deux points sont à égale distance des extrémités d'un segment, alors la droite qui passe par ces deux points est la médiatrice de ce segment.

Quadrilatères et parallélogrammes

10.1 Si un quadrilatère est un parallélogramme, alors ses côtés opposés sont parallèles

10.2 Si un quadrilatère a ses côtés opposés parallèles, alors c'est un parallélogramme

11.1 Si un quadrilatère est un parallélogramme, alors le milieu d'une de ses diagonales est aussi le milieu de l'autre diagonale

11.2 Dans un quadrilatère, si le milieu d'une diagonale est aussi le milieu de l'autre diagonale alors ce quadrilatère est un parallélogramme

11.3 Si un quadrilatère est un parallélogramme, alors le point d'intersection des diagonales est le milieu des diagonales

12.1 Si un quadrilatère est un parallélogramme, alors ses côtés opposés sont égaux

12.2 Si un quadrilatère a ses côtés opposés égaux, alors c'est un parallélogramme

13.1 Si un quadrilatère a deux côtés opposés égaux et parallèles, alors c'est un parallélogramme

14.1 Si un parallélogramme a deux côtés consécutifs perpendiculaires, alors c'est un rectangle

15.1 Si un parallélogramme a ses diagonales de même longueur, alors c'est un rectangle

16.1 Si un quadrilatère a ses côtés consécutifs perpendiculaires deux à deux, alors c'est un rectangle

17.1 Si un parallélogramme a deux côtés consécutifs égaux, alors c'est un losange

18.1 Si un parallélogramme a ses diagonales perpendiculaires, alors c'est un losange

19.1 Si un quadrilatère a ses côtés consécutifs égaux deux à deux, alors c'est un losange

Triangles et droite des milieux

20.1 La droite passant par le milieu du côté d'un triangle, et parallèle à un autre côté, passe par le milieu du troisième côté

21.1 Dans un triangle, la droite qui passe par les milieux de deux côtés est parallèle au troisième côté

21.2 Dans un triangle, le segment qui joint les milieux de deux côtés vaut la moitié du troisième côté

22.1 Dans un triangle rectangle, la médiane issue de l'angle droit vaut la moitié de l'hypoténuse

23.1 Si un triangle est rectangle, alors il a deux côtés perpendiculaires

23.2 Si un triangle a deux côtés perpendiculaires, alors c'est un triangle rectangle

24.1 Si un point X appartient à un cercle de diamètre [YZ], alors le triangle XYZ est rectangle en X

24.2 Si le triangle XYZ est rectangle en X, alors X appartient au cercle de diamètre [YZ]

Théorèmes implicites

- A,B,C alignés, A,B,D alignés $\Rightarrow$ A,C,D alignés, B,C,D alignés;
- I milieu de [AB] $\Rightarrow$ I,A,B alignés;
- [AB] diamètre de C $\Rightarrow$ A $\in$ C, B $\in$ C;
- ABCD losange $\Rightarrow$ ABCD parallélogramme;
- ABCD losange $\Rightarrow$ AB = BC, BC = CD, CD = DA, DA = AB
- ABCD losange $\Rightarrow$ (AC) $\perp$ (BD)
- ABCD rectangle $\Rightarrow$ ABCD parallélogramme;
- ABCD rectangle $\Rightarrow$ (AB) $\perp$ (BC), (BC) $\perp$ (CD), (CD) $\perp$ (DA), (AD) $\perp$ (AB)
- ABCD rectangle $\Rightarrow$ AC = BD

Annexe C

Base d'exercices

A1 Soit ABC un triangle, et M le milieu de [AB]. K est le point tel que M est le milieu de [CK]. Montre que ACBK est un parallélogramme.

A2 Soit ABC un triangle, M le milieu de [BC] et H le milieu de [AM]. La parallèle à (BC) passant par H coupe (AB) en K. Montre que K est le milieu de [AB].

A3 Soit ABC un triangle. I est un point du segment [AB] et J un point du segment [AC] tels que (IJ) est parallèle à (BC). H est le pied de la hauteur issue de A dans AIJ. K est le pied de la hauteur issue de A dans ABC. Montrer que A, H et K sont alignés.

A4 Soit ABC un triangle. I est le milieu de [AB], et J est le milieu de [AC]. La parallèle à (AB) passant par J coupe (BC) en K. Montre que BIJK est un parallélogramme.

A5 On considère le parallélogramme ABCD et on appelle I le point d'intersection de [AC] et de [BD]. Soient F et G les milieux de [BC] et de [DC], et M le point d'intersection de [IC] et [FG]. Démontrer que M est le milieu de [IC].

A6 Soit ABC un triangle, et H le pied de la hauteur issue de A. La parallèle à (BC) passant par A coupe la parallèle à (AH) passant par B en un point K. Montre que AHBK est un rectangle.

A7 Soit ABC un triangle isocèle en A. La parallèle à (AB) passant par C coupe la parallèle à (AC) passant par B en un point K. Montre que ABKC est un losange.

A8 Soit ABC un triangle rectangle en A. M est le milieu de [BC], et D est le milieu de [AB]. H est le pied de la hauteur issue de A dans ABC. G est le cercle de diamètre [AM]. Démontre que H appartient au cercle G.

B1 ABCD est un parallélogramme. E est le point tel que ABEC est un parallélogramme. Montre que C, D, E sont alignés.

B2 Soit ABC un triangle. T et R sont deux points situés sur le segment [AC], tels que AR = RT et RT = TC. Le point P est tel que B est le milieu de [CP]. La droite (PR) coupe la droite (AB) en M. Montre que M est le milieu de [AB].

B3 Soit ABC un triangle. I est le milieu de [AC], et J celui de [AB]. La parallèle à (BC) passant par A coupe (BI) en D. Montre que ABCD est un parallélogramme.

B4 ABC est un triangle. D est le milieu de [BC], E celui de [AC] et F celui de [AB]. I est le point d'intersection de (AD) et (FE). Démontre que I est le milieu de [EF].

B5 ABC est un triangle isocèle en A, et M est le milieu de [BC]. Montre que MAB est un triangle rectangle en M.

B6 Soit ABCD un parallélogramme. La parallèle à (BD) passant par A coupe (CD) en I. La parallèle à (BD) passant par C coupe (AB) en J. Montre que AICJ est un parallélogramme.

B7 Soient A, B, C, D, E, F six points placés de telle sorte que ABCD, BCED et ACFD soient des parallélogrammes. Le point O est le milieu de [CD]. Montre que ABFE est un parallélogramme.

B8 Soit ABC un triangle. M est le milieu de [BC]. H est le point de [AM] tel que (BH) est perpendiculaire à (AM). K est le point tel que M est le milieu de [HK]. Montre que KMC est un triangle rectangle en K.

B9 Soit ABC un triangle et U le milieu de [AB]. La parallèle à (BC) passant par U coupe (AC) en H. La parallèle à (AC) passant par U coupe (BC) en I. Montrer que (AB) est parallèle à (HI).

B10 Soit ABCD un quadrilatère. I est le milieu de [AB]. La parallèle à (BC) passant par I coupe (AC) en J. La parallèle à (DC) passant par J coupe (AD) en K. Montrer que (IK) est parallèle à (BD).

B11 ABC est un triangle. D, E, F sont les milieux respectifs de [BC], [AC], [AB]. I et J sont les milieux de [DE] et [DF]. Montre que (IJ) est parallèle à (BC).

B12 Soit ABC un triangle rectangle en A. Le point N est le milieu de l'hypoténuse. Le point M est le milieu de [AB]. Montre que (MN) est perpendiculaire à (AB).

B13 Soit ABC un triangle. H est un point quelconque de [BC]. D, E, F sont les milieux respectifs de [AB], [AH], [AC]. Démontre que D, E, F sont alignés.

B14 Soit ABCD un trapèze de bases [AB] et [CD]. I, J, K sont les milieux respectifs de [AD], [BC], [BD]. Démontre que I, J, K sont alignés.

B15 Dessine un triangle EBC rectangle en E, et appelle M le milieu de l'hypoténuse. Dessine le point A tel que M soit le milieu de [AE]. Démontre que le quadrilatère ACEB est un rectangle.

B16 Soit C un cercle de diamètre [AB]. P est un point quelconque de C. I et J sont tels que P est le milieu de [AI] et de [BJ]. Démontre que ABIJ est un losange.

C1 Soit OCL un triangle. I est le milieu de [CL]. D est le point tel que ODIL est un parallélogramme. M est le milieu de [ID]. Montre que M est le milieu de [OC].

C2 Soit FURS un parallélogramme. I est le milieu de [UR]. G est le point tel que I est le milieu de [FG]. Montre que R est le milieu de [GS].

C3 Soit ABC un triangle. I est le milieu de [AB], et J celui de [AC]. D et E sont tels que I est le milieu de [CD], J est le milieu de [BE]. Montre que A est le milieu de [DE].

C4 Soit ABC un triangle rectangle en A, et M le milieu de [BC]. I est le milieu de [AB], J celui de [AC]. Montre que AIMJ est un rectangle.

C5 Soit ABCD un parallélogramme de centre O. La parallèle à (BD) passant par A coupe (CD) en I. La parallèle à (BD) passant par C coupe (AB) en J. Montre que BIDJ est un parallélogramme.

C6 Soit ABC un triangle rectangle en A. M est le milieu de [BC], et D est le milieu de [AB]. H est le pied de la hauteur issue de A dans ABC. G est le cercle de diamètre [AM]. Démontre que D appartient au cercle G.

C7 Soit OAB un triangle rectangle en O. Le cercle C, de diamètre [AO], recoupe le cercle D, de diamètre [OB], en un point M. Démontre que A, M, B sont alignés.

C8 Soit ABC un triangle quelconque. I est le milieu de [AB]. J est le milieu de [AC]. Les droites (IC) et (JB) se coupent en un point O. Les points M et N sont les milieux respectifs de [OB] et [OC]. Montre que IJNM est un parallélogramme.

C9 Soit ABCD un quadrilatère quelconque. Les points I, J, K, et L sont les milieux respectifs de [AD], [AB], [BC] et [CD]. Montre que IJKL est un parallélogramme.

C10 Soit C un cercle de centre O et de diamètre [AB]. D est le cercle de diamètre [AO]. Q est un point de C, tel que (QA) coupe le cercle D en un point P. Démontre que P est le milieu de [AQ].